# The Divergence of the Prime Harmonic Series

Manuel Eberl

April 14, 2026

**Abstract**

In this work, we prove the lower bound $\ln(H_n) - \ln(\frac{5}{3})$ for the partial sum of the Prime Harmonic series and, based on this, the divergence of the Prime Harmonic Series $\sum_{p=1}^{n} [p \text{ prime}] \cdot \frac{1}{p}$. The proof relies on the unique squarefree decomposition of natural numbers. This proof is similar to Euler's original proof (which was highly informal and morally questionable). Its advantage over proofs by contradiction, like the famous one by Paul Erdős, is that it provides a relatively good lower bound for the partial sums.

## Contents



## 1 Auxiliary lemmas

**theory** *Prime-Harmonic-Misc*
**imports**
*Complex-Main*
*HOL−Number-Theory.Number-Theory*
**begin**

**lemma** *sum-list-nonneg*: $\forall x \in set\ xs.\ x \geq 0 \Longrightarrow sum\text{-}list\ xs \geq (0 :: 'a :: ordered\text{-}ab\text{-}group\text{-}add)$
**by** (*induction xs*) *auto*

**lemma** *sum-telescope′*:
**assumes** $m \leq n$
**shows** $(\sum k = Suc\ m..n.\ f\ k - f\ (Suc\ k)) = f\ (Suc\ m) - (f\ (Suc\ n) :: 'a :: ab\text{-}group\text{-}add)$

```
  by (rule dec-induct[OF assms]) (simp-all add: algebra-simps)

lemma dvd-prodI:
  assumes finite A x ∈ A
  shows   f x dvd prod f A
proof −
  from assms have prod f A = f x ∗ prod f (A − {x})
    by (intro prod.remove) simp-all
  thus ?thesis by simp
qed

lemma dvd-prodD: finite A =⇒ prod f A dvd x =⇒ a ∈ A =⇒ f a dvd x
  by (erule dvd-trans[OF dvd-prodI])

lemma multiplicity-power-nat:
  prime p =⇒ n > 0 =⇒ multiplicity p (n ^ k :: nat) = k ∗ multiplicity p n
  by (induction k) (simp-all add: prime-elem-multiplicity-mult-distrib)

lemma multiplicity-prod-prime-powers-nat':
  finite S =⇒ ∀ p∈S. prime p =⇒ prime p =⇒
     multiplicity p (∏ S :: nat) = (if p ∈ S then 1 else 0)
  using multiplicity-prod-prime-powers[of S p λ-. 1] by simp

lemma prod-prime-subset:
  assumes finite A finite B
  assumes ⋀x. x ∈ A =⇒ prime (x::nat)
  assumes ⋀x. x ∈ B =⇒ prime x
  assumes ∏ A dvd ∏ B
  shows   A ⊆ B
proof
  fix x assume x: x ∈ A
  from assms(4)[of 0] have 0 ∉ B by auto
  with assms have nonzero: ∀ z∈B. z ≠ 0 by (intro ballI notI) auto

  from x assms have 1 = multiplicity x (∏ A)
    by (subst multiplicity-prod-prime-powers-nat') simp-all
 also from assms nonzero have . . . ≤ multiplicity x (∏ B) by (intro dvd-imp-multiplicity-le)
auto
  finally have multiplicity x (∏ B) > 0 by simp
  moreover from assms x have prime x by simp
  ultimately show x ∈ B using assms(2,4)
   by (subst (asm) multiplicity-prod-prime-powers-nat') (simp-all split: if-split-asm)
qed

lemma prod-prime-eq:
  assumes finite A finite B ⋀x. x ∈ A =⇒ prime (x::nat) ⋀x. x ∈ B =⇒ prime
x ∏ A = ∏ B
  shows   A = B
  using assms by (intro equalityI prod-prime-subset) simp-all
```

**lemma** *ln-ln-nonneg*:
  **assumes** *x*: *x* ≥ (*3* :: *real*)
  **shows** *ln* (*ln x*) ≥ *0*
**proof** −
  **have** *exp 1* ≤ (*3*::*real*) **by** (*rule exp-le*)
  **hence** *ln* (*exp 1*) ≤ *ln* (*3* :: *real*) **by** (*subst ln-le-cancel-iff*) *simp-all*
  **also from** *x* **have** . . . ≤ *ln x* **by** (*subst ln-le-cancel-iff*) *simp-all*
  **finally have** *ln 1* ≤ *ln* (*ln x*) **using** *x* **by** (*subst ln-le-cancel-iff*) *simp-all*
  **thus** *?thesis* **by** *simp*
**qed**

**end**

# 2 Squarefree decomposition of natural numbers

**theory** *Squarefree-Nat*
**imports**
  *Main*
  *HOL−Number-Theory.Number-Theory*
  *Prime-Harmonic-Misc*
**begin**

The squarefree part of a natural number is the set of all prime factors that appear with odd multiplicity. The square part, correspondingly, is what remains after dividing by the squarefree part.

**definition** *squarefree-part* :: *nat* ⇒ *nat set* **where**
  *squarefree-part n* = {*p*∈*prime-factors n*. *odd* (*multiplicity p n*)}

**definition** *square-part* :: *nat* ⇒ *nat* **where**
  *square-part n* = (*if n* = *0 then 0 else* (∏ *p*∈*prime-factors n*. *p* ^ (*multiplicity p n div 2*)))

**lemma** *squarefree-part-0* [*simp*]: *squarefree-part 0* = {}
  **by** (*simp add*: *squarefree-part-def*)

**lemma** *square-part-0* [*simp*]: *square-part 0* = *0*
  **by** (*simp add*: *square-part-def*)

**lemma** *squarefree-decompose*: ∏ (*squarefree-part n*) ∗ *square-part n* ^ *2* = *n*
**proof** (*cases n* = *0*)
  **case** *False*
  **define** *A s* **where** *A* = *squarefree-part n* **and** *s* = *square-part n*
  **have** (∏ *A*) = (∏ *p*∈*A*. *p* ^ (*multiplicity p n mod 2*))
    **by** (*intro prod.cong*) (*auto simp*: *A-def squarefree-part-def elim*!: *oddE*)
  **also have** . . . = (∏ *p*∈*prime-factors n*. *p* ^ (*multiplicity p n mod 2*))
    **by** (*intro prod.mono-neutral-left*) (*auto simp*: *A-def squarefree-part-def*)
  **also from** *False* **have** . . . ∗ *s*^*2* = *n*

**by** (*simp add*: *s-def square-part-def prod.distrib* [*symmetric*] *power-add* [*symmetric*]
*power-mult* [*symmetric*] *prime-factorization-nat* [*symmetric*]
*algebra-simps*
*prod-power-distrib*)
**finally show** $\prod A * s\hat{}2 = n$ **.**
**qed** *simp*

**lemma** *squarefree-part-pos* [*simp*]: $\prod$(*squarefree-part n*) > *0*
**using** *prime-gt-0-nat* **unfolding** *squarefree-part-def* **by** *auto*

**lemma** *squarefree-part-ge-Suc-0* [*simp*]: $\prod$(*squarefree-part n*) ≥ *Suc 0*
**using** *squarefree-part-pos*[*of n*] **by** *presburger*

**lemma** *squarefree-part-subset* [*intro*]: *squarefree-part n* ⊆ *prime-factors n*
**unfolding** *squarefree-part-def* **by** *auto*

**lemma** *squarefree-part-finite* [*simp*]: *finite* (*squarefree-part n*)
**by** (*rule finite-subset*[*OF squarefree-part-subset*]) *simp*

**lemma** *squarefree-part-dvd* [*simp*]: $\prod$(*squarefree-part n*) *dvd n*
**by** (*subst* (*2*) *squarefree-decompose* [*of n*, *symmetric*]) *simp*

**lemma** *squarefree-part-dvd'* [*simp*]: *p* ∈ *squarefree-part n* ⟹ *p dvd n*
**by** (*rule dvd-prodD*[*OF - squarefree-part-dvd*]) *simp-all*

**lemma** *square-part-dvd* [*simp*]: *square-part n* ^ *2 dvd n*
**by** (*subst* (*2*) *squarefree-decompose* [*of n*, *symmetric*]) *simp*

**lemma** *square-part-dvd'* [*simp*]: *square-part n dvd n*
**by** (*subst* (*2*) *squarefree-decompose* [*of n*, *symmetric*]) *simp*

**lemma** *squarefree-part-le*: *p* ∈ *squarefree-part n* ⟹ *p* ≤ *n*
**by** (*cases n* = *0*) (*simp-all add*: *dvd-imp-le*)

**lemma** *square-part-le*: *square-part n* ≤ *n*
**by** (*cases n* = *0*) (*simp-all add*: *dvd-imp-le*)

**lemma** *square-part-le-sqrt*: *square-part n* ≤ *nat* ⌊*sqrt* (*real n*)⌋
**proof** −
**have** *1* ∗ *square-part n* ^ *2* ≤ $\prod$(*squarefree-part n*) ∗ *square-part n* ^ *2*
**by** (*intro mult-right-mono*) *simp-all*
**also have** ... = *n* **by** (*rule squarefree-decompose*)
**finally have** *real* (*square-part n* ^ *2*) ≤ *real n* **by** (*subst of-nat-le-iff*) *simp*
**hence** *sqrt* (*real* (*square-part n* ^ *2*)) ≤ *sqrt* (*real n*)
**by** (*subst real-sqrt-le-iff*) *simp-all*
**also have** *sqrt* (*real* (*square-part n* ^ *2*)) = *real* (*square-part n*) **by** *simp*
**finally show** *?thesis* **by** *linarith*
**qed**

**lemma** *square-part-pos* [*simp*]: $n > 0 \Longrightarrow$ *square-part* $n > 0$
  **unfolding** *square-part-def* **using** *prime-gt-0-nat* **by** *auto*

**lemma** *square-part-ge-Suc-0* [*simp*]: $n > 0 \Longrightarrow$ *square-part* $n \geq$ *Suc 0*
  **using** *square-part-pos*[*of n*] **by** *presburger*

**lemma** *zero-not-in-squarefree-part* [*simp*]: $0 \notin$ *squarefree-part n*
  **unfolding** *squarefree-part-def* **by** *auto*

**lemma** *multiplicity-squarefree-part*:
  *prime p* $\Longrightarrow$ *multiplicity p* ($\prod$(*squarefree-part n*)) = (*if p* $\in$ *squarefree-part n then 1 else 0*)
  **using** *squarefree-part-subset*[*of n*]
  **by** (*intro multiplicity-prod-prime-powers-nat'*) *auto*

The squarefree part really is square, its only square divisor is 1.

**lemma** *square-dvd-squarefree-part-iff*:
  $x\hat{}2$ *dvd* $\prod$(*squarefree-part n*) $\longleftrightarrow x = 1$
**proof** (*rule iffI*, *rule multiplicity-eq-nat*)
  **assume** *dvd*: $x\hat{}2$ *dvd* $\prod$(*squarefree-part n*)
  **hence** $x \neq 0$ **using** *squarefree-part-subset*[*of n*] *prime-gt-0-nat* **by** (*intro notI*) *auto*
  **thus** *x*: $x > 0$ **by** *simp*

  **fix** *p* :: *nat* **assume** *p*: *prime p*
  **from** *p x* **have** $2 *$ *multiplicity p x* = *multiplicity p* ($x\hat{}2$)
    **by** (*simp add*: *multiplicity-power-nat*)
  **also from** *dvd* **have** ... $\leq$ *multiplicity p* ($\prod$(*squarefree-part n*))
    **by** (*intro dvd-imp-multiplicity-le*) *simp-all*
  **also have** ... $\leq 1$ **using** *multiplicity-squarefree-part*[*of p n*] *p* **by** *simp*
  **finally show** *multiplicity p x* = *multiplicity p 1* **by** *simp*
**qed** *simp-all*

**lemma** *squarefree-decomposition-unique1*:
  **assumes** *squarefree-part m* = *squarefree-part n*
  **assumes** *square-part m* = *square-part n*
  **shows** $m = n$
  **by** (*subst* (*1 2*) *squarefree-decompose* [*symmetric*]) (*simp add*: *assms*)

**lemma** *squarefree-decomposition-unique2*:
  **assumes** *n*: $n > 0$
  **assumes** *decomp*: $n = (\prod A2 * s2\hat{}2)$
  **assumes** *prime*: $\bigwedge x.\ x \in A2 \Longrightarrow$ *prime x*
  **assumes** *fin*: *finite A2*
  **assumes** *s2-nonneg*: $s2 \geq 0$
  **shows** $A2$ = *squarefree-part n* **and** $s2$ = *square-part n*
**proof** −

```
  define A1 s1 where A1 = squarefree-part n and s1 = square-part n
  have finite A1 unfolding A1-def by simp
  note fin = ‹finite A1› ‹finite A2›

  have A1 ⊆ prime-factors n unfolding A1-def using squarefree-part-subset .
  note subset = this prime

  have ∏A1 > 0 ∏A2 > 0 using subset(1)  prime-gt-0-nat
    by (auto intro!: prod-pos dest: prime)
  from n have s1 > 0 unfolding s1-def by simp
  from assms have s2 ≠ 0 by (intro notI) simp
  hence s2 > 0 by simp
  note pos = ‹∏A1 > 0› ‹∏A2 > 0› ‹s1 > 0› ‹s2 > 0›

  have eq': multiplicity p s1 = multiplicity p s2
          multiplicity p (∏A1) = multiplicity p (∏A2)
    if  p: prime p for p
  proof −
    define m where m = multiplicity p
    from decomp have m (∏A1 ∗ s1^2) = m (∏A2 ∗ s2^2) unfolding A1-def
s1-def
      by (simp add: A1-def s1-def squarefree-decompose)
    with p pos have eq: m (∏A1) + 2 ∗ m s1 = m (∏A2) + 2 ∗ m s2
     by (simp add: m-def prime-elem-multiplicity-mult-distrib multiplicity-power-nat)
    moreover from fin subset p have m (∏A1) ≤ 1 m (∏A2) ≤ 1 unfolding
m-def
      by ((subst multiplicity-prod-prime-powers-nat', auto)[])+
    ultimately show m s1 = m s2 by linarith
    with eq show m (∏A1) = m (∏A2) by simp
  qed

  show s2 = square-part n
    by (rule multiplicity-eq-nat) (insert pos eq'(1), auto simp: s1-def)
  have ∏A2 = ∏(squarefree-part n)
    by (rule multiplicity-eq-nat) (insert pos eq'(2), auto simp: A1-def)
  with fin subset show A2 = squarefree-part n unfolding A1-def
    by (intro prod-prime-eq) auto
qed

lemma squarefree-decomposition-unique2':
  assumes decomp: (∏A1 ∗ s1^2) = (∏A2 ∗ s2^2 :: nat)
  assumes fin: finite A1 finite A2
  assumes subset: ⋀x. x ∈ A1 ⟹ prime x ⋀x. x ∈ A2 ⟹ prime x
  assumes pos: s1 > 0 s2 > 0
  defines n ≡ ∏A1 ∗ s1^2
  shows A1 = A2 s1 = s2
proof −
  from pos have n: n > 0 using prime-gt-0-nat
    by (auto simp: n-def intro!: prod-pos dest: subset)
```

**have** *A1 = squarefree-part n s1 = square-part n*
**by** ((*rule squarefree-decomposition-unique2*[*of n A1 s1*], *insert assms n*, *simp-all*)[])+
**moreover have** *A2 = squarefree-part n s2 = square-part n*
**by** ((*rule squarefree-decomposition-unique2*[*of n A2 s2*], *insert assms n*, *simp-all*)[])+
**ultimately show** *A1 = A2 s1 = s2* **by** *simp-all*
**qed**

The following is a nice and simple lower bound on the number of prime numbers less than a given number due to Erdős. In particular, it implies that there are infinitely many primes.

**lemma** *primes-lower-bound*:
**fixes** *n* :: *nat*
**assumes** *n* > *0*
**defines** $\pi \equiv \lambda n.$ *card* {*p. prime p* ∧ *p* ≤ *n*}
**shows** *real* ($\pi$ *n*) ≥ *ln* (*real n*) / *ln 4*
**proof** −
**have** *real n* = *real* (*card* {*1..n*}) **by** *simp*
**also have** {*1..n*} = (λ(*A*, *b*). ∏ *A* ∗ *b*^*2*) ‘ (λ*n*. (*squarefree-part n*, *square-part n*)) ‘ {*1..n*}
**unfolding** *image-comp o-def squarefree-decompose case-prod-unfold fst-conv snd-conv* **by** *simp*
**also have** *card* ... ≤ *card* ((λ*n*. (*squarefree-part n*, *square-part n*)) ‘ {*1..n*})
**by** (*rule card-image-le*) *simp-all*
**also have** ... ≤ *card* (*squarefree-part* ‘ {*1..n*} × *square-part* ‘ {*1..n*})
**by** (*rule card-mono*) *auto*
**also have** *real* ... = *real* (*card* (*squarefree-part* ‘ {*1..n*})) ∗ *real* (*card* (*square-part* ‘ {*1..n*}))
**by** *simp*
**also have** ... ≤ *2* ^ π *n* ∗ *sqrt* (*real n*)
**proof** (*rule mult-mono*)
**have** *card* (*squarefree-part* ‘ {*1..n*}) ≤ *card* (*Pow* {*p. prime p* ∧ *p* ≤ *n*})
**using** *squarefree-part-subset squarefree-part-le* **by** (*intro card-mono*) *force*+
**also have** *real* ... = *2* ^ π *n* **by** (*simp add*: π*-def card-Pow*)
**finally show** *real* (*card* (*squarefree-part* ‘ {*1..n*})) ≤ *2* ^ π *n* **by** − *simp-all*
**next**
**have** *square-part k* ≤ *nat* ⌊*sqrt n*⌋ **if** *k* ≤ *n* **for** *k*
**by** (*rule order.trans*[*OF square-part-le-sqrt*])
(*insert that*, *auto intro*!: *nat-mono floor-mono*)
**hence** *card* (*square-part* ‘ {*1..n*}) ≤ *card* {*1..nat* ⌊*sqrt n*⌋}
**by** (*intro card-mono*) (*auto intro*: *order.trans*[*OF square-part-le-sqrt*])
**also have** ... = *nat* ⌊*sqrt n*⌋ **by** *simp*
**also have** *real* ... ≤ *sqrt n* **by** *simp*
**finally show** *real* (*card* (*square-part* ‘ {*1..n*})) ≤ *sqrt* (*real n*) **by** − *simp-all*
**qed** *simp-all*
**finally have** *real n* ≤ *2* ^ π *n* ∗ *sqrt* (*real n*) **by** − *simp-all*
**with** ‹*n* > *0*› **have** *ln* (*real n*) ≤ *ln* (*2* ^ π *n* ∗ *sqrt* (*real n*))
**by** (*subst ln-le-cancel-iff*) *simp-all*
**moreover have** *ln* (*4* :: *real*) = *real 2* ∗ *ln 2* **by** (*subst ln-realpow* [*symmetric*]) *simp-all*

**ultimately show** *?thesis* **using** ‹*n* > *0*›
  **by** (*simp add*: *ln-mult ln-realpow*[*of* - *π n*] *ln-sqrt field-simps*)
**qed**

**end**

# 3 The Prime Harmonic Series

**theory** *Prime-Harmonic*
**imports**
  *HOL−Analysis.Analysis*
  *HOL−Number-Theory.Number-Theory*
  *Prime-Harmonic-Misc*
  *Squarefree-Nat*
**begin**

## 3.1 Auxiliary equalities and inequalities

First of all, we prove the following result about rearranging a product over a set into a sum over all subsets of that set.

**lemma** *prime-harmonic-aux1*:
  **fixes** *A* :: *'a* :: *field set*
  **shows** *finite A* ⟹ (∏*x*∈*A*. *1* + *1* / *x*) = (∑*x*∈*Pow A*. *1* / ∏*x*)
**proof** (*induction rule*: *finite-induct*)
  **fix** *a* :: *'a* **and** *A* :: *'a set*
  **assume** *a*: *a* ∉ *A* **and** *fin*: *finite A*
  **assume** *IH*: (∏*x*∈*A*. *1* + *1* / *x*) = (∑*x*∈*Pow A*. *1* / ∏*x*)
  **from** *a* **and** *fin* **have** (∏*x*∈*insert a A*. *1* + *1* / *x*) = (*1* + *1* / *a*) ∗ (∏*x*∈*A*. *1* + *1* / *x*) **by** *simp*
  **also from** *fin* **have** ... = (∑*x*∈*Pow A*. *1* / ∏*x*) + (∑*x*∈*Pow A*. *1* / (*a* ∗ ∏*x*))
    **by** (*subst IH*) (*auto simp add*: *algebra-simps sum-divide-distrib*)
  **also from** *fin a* **have** (∑*x*∈*Pow A*. *1* / (*a* ∗ ∏*x*)) = (∑*x*∈*Pow A*. *1* / ∏(*insert a x*))
    **by** (*intro sum.cong refl*, *subst prod.insert*) (*auto dest*: *finite-subset*)
  **also from** *a* **have** ... = (∑*x*∈*insert a* ‘ *Pow A*. *1* / ∏*x*)
    **by** (*subst sum.reindex*) (*auto simp*: *inj-on-def*)
  **also from** *fin a* **have** (∑*x*∈*Pow A*. *1* / ∏*x*) + ... = (∑*x*∈*Pow A* ∪ *insert a* ‘ *Pow A*. *1* / ∏*x*)
    **by** (*intro sum.union-disjoint* [*symmetric*]) (*simp*, *simp*, *blast*)
  **also have** *Pow A* ∪ *insert a* ‘ *Pow A* = *Pow* (*insert a A*) **by** (*simp only*: *Pow-insert*)
  **finally show** (∏*x*∈*insert a A*. *1* + *1* / *x*) = (∑*x*∈*Pow* (*insert a A*). *1* / ∏*x*) .
**qed** *simp*

Next, we prove a simple and reasonably accurate upper bound for the sum of the squares of any subset of the natural numbers, derived by simple telescoping. Our upper bound is approximately 1.67; the exact value is $\frac{\pi^2}{6} \approx 1.64$. (cf. Basel problem)

**lemma** *prime-harmonic-aux2*:
  **assumes** *finite* (*A* :: *nat set*)
  **shows** ($\sum$ *k*∈*A*. *1* / (*real k* ^ *2*)) ≤ *5/3*
**proof** −
  **define** *n* **where** *n* = *max 2* (*Max A*)
  **have** *n*: *n* ≥ *Max A n* ≥ *2* **by** (*auto simp*: *n-def*)
  **with** *assms* **have** *A* ⊆ {*0..n*} **by** (*auto intro*: *order.trans*[*OF Max-ge*])
  **hence** ($\sum$ *k*∈*A*. *1* / (*real k* ^ *2*)) ≤ ($\sum$ *k*=*0..n*. *1* / (*real k* ^ *2*)) **by** (*intro sum-mono2*) *auto*
  **also from** *n* **have** ... = *1* + ($\sum$ *k*=*Suc 1..n*. *1* / (*real k* ^ *2*)) **by** (*simp add*: *sum.atLeast-Suc-atMost*)
  **also have** ($\sum$ *k*=*Suc 1..n*. *1* / (*real k* ^ *2*)) ≤
      ($\sum$ *k*=*Suc 1..n*. *1* / (*real k* ^ *2* − *1/4*)) **unfolding** *power2-eq-square*
    **by** (*intro sum-mono divide-left-mono mult-pos-pos*)
      (*linarith*, *simp-all add*: *field-simps less-1-mult*)
  **also have** ... = ($\sum$ *k*=*Suc 1..n*. *1* / (*real k* − *1/2*) − *1* / (*real* (*Suc k*) − *1/2*))
    **by** (*intro sum.cong refl*) (*simp-all add*: *field-simps power2-eq-square*)
  **also from** *n* **have** ... = *2* / *3* − *1* / (*1* / *2* + *real n*)
    **by** (*subst sum-telescope′*) *simp-all*
  **also have** *1* + ... ≤ *5/3* **by** *simp*
  **finally show** *?thesis* **by** − *simp*
**qed**

## 3.2 Estimating the partial sums of the Prime Harmonic Series

We are now ready to show our main result: the value of the partial prime harmonic sum over all primes no greater than $n$ is bounded from below by the $n$-th harmonic number $H_n$ minus some constant.

In our case, this constant will be $\frac{5}{3}$. As mentioned before, using a proof of the Basel problem can improve this to $\frac{\pi^2}{6}$, but the improvement is very small and the proof of the Basel problem is a very complex one.

The exact asymptotic behaviour of the partial sums is actually $\ln(\ln n) + M$, where $M$ is the Meissel–Mertens constant (approximately 0.261).

**theorem** *prime-harmonic-lower*:
  **assumes** *n*: *n* ≥ *2*
  **shows** ($\sum$ *p*←*primes-upto n*. *1* / *real p*) ≥ *ln* (*harm n*) − *ln* (*5/3*)
**proof** −
  — the set of primes that we will allow in the squarefree part
  **define** *P* **where** *P n* = *set* (*primes-upto n*) **for** *n*
  **{**
    **fix** *n* :: *nat*
    **have** *finite* (*P n*) **by** (*simp add*: *P-def*)
  **}** **note** [*simp*] = *this*

— The function that combines the squarefree part and the square part
**define** *f* **where** *f* = (λ(*R*, *s* :: *nat*). ∏ *R* ∗ *s*^*2*)

— *f* is injective if the squarefree part contains only primes and the square part is positive.
**have** *inj*: *inj-on f* (*Pow* (*P n*)×{*1*..*n*})
**proof** (*rule inj-onI*, *clarify*, *rule conjI*)
**fix** *A1 A2* :: *nat set* **and** *s1 s2* :: *nat*
**assume** *A*: *A1* ⊆ *P n A2* ⊆ *P n s1* ∈ {*1*..*n*} *s2* ∈ {*1*..*n*} *f* (*A1*, *s1*) = *f* (*A2*, *s2*)
**have** *fin*: *finite A1 finite A2* **by** (*rule A*(*1*,*2*)[*THEN finite-subset*], *simp*)+
**show** *A1* = *A2 s1* = *s2*
**by** ((*rule squarefree-decomposition-unique2* ′[*of A1 s1 A2 s2*],
*insert A fin*, *auto simp*: *f-def P-def set-primes-upto*)[])+
**qed**

— *f* hits every number between *1* and *n*. It also hits a lot of other numbers, but we do not care about those, since we only need a lower bound.
**have** *surj*: {*1*..*n*} ⊆ *f* ‘ (*Pow* (*P n*)×{*1*..*n*})
**proof**
**fix** *x* **assume** *x*: *x* ∈ {*1*..*n*}
**have** *x* = *f* (*squarefree-part x*, *square-part x*) **by** (*simp add*: *f-def square-free-decompose*)
**moreover have** *squarefree-part x* ∈ *Pow* (*P n*) **using** *squarefree-part-subset*[*of x*] *x*
**by** (*auto simp*: *P-def set-primes-upto intro*: *order*.*trans*[*OF squarefree-part-le*[*of - x*]])
**moreover have** *square-part x* ∈ {*1*..*n*} **using** *x*
**by** (*auto simp*: *Suc-le-eq intro*: *order*.*trans*[*OF square-part-le*[*of x*]])
**ultimately show** *x* ∈ *f* ‘ (*Pow* (*P n*)×{*1*..*n*}) **by** *simp*
**qed**

— We now show the main result by rearranging the sum over all primes to a product over all all squarefree parts times a sum over all square parts, and then applying some simple-minded approximation
**have** *harm n* = (∑ *n*=*1*..*n*. *1* / *real n*) **by** (*simp add*: *harm-def field-simps*)
**also from** *surj* **have** ... ≤ (∑ *n*∈*f* ‘ (*Pow* (*P n*)×{*1*..*n*}). *1* / *real n*)
**by** (*intro sum-mono2 finite-imageI finite-cartesian-product*) *simp-all*
**also from** *inj* **have** ... = (∑ *x*∈*Pow* (*P n*)×{*1*..*n*}. *1* / *real* (*f x*))
**by** (*subst sum*.*reindex*) *simp-all*
**also have** ... = (∑ *A*∈*Pow* (*P n*). *1* / *real* (∏ *A*)) ∗ (∑ *k*=*1*..*n*. *1* / (*real k*)^*2*)
**unfolding** *f-def*
**by** (*subst sum-product*, *subst sum*.*cartesian-product*) (*simp add*: *case-prod-beta*)
**also have** ... ≤ (∑ *A*∈*Pow* (*P n*). *1* / *real* (∏ *A*)) ∗ (*5*/*3*)
**by** (*intro mult-left-mono prime-harmonic-aux2 sum-nonneg*)
(*auto simp*: *P-def intro*!: *prod-nonneg*)
**also have** (∑ *A*∈*Pow* (*P n*). *1* / *real* (∏ *A*)) = (∑ *A*∈((‘) *real*) ‘ *Pow* (*P n*). *1* / ∏ *A*)
**by** (*subst sum*.*reindex*) (*auto simp*: *inj-on-def inj-image-eq-iff prod*.*reindex*)

**also have** $((\text{`}) \; real) \; \text{`} \; Pow \; (P \; n) = Pow \; (real \; \text{`} \; P \; n)$ **by** (*intro image-Pow-surj refl*)
**also have** $(\sum A{\in}Pow \; (real \; \text{`} \; P \; n). \; 1 \; / \; \prod A) = (\prod x{\in}real \; \text{`} \; P \; n. \; 1 + 1 \; / \; x)$
**by** (*intro prime-harmonic-aux1* [*symmetric*] *finite-imageI*) *simp-all*
**also have** $\ldots = (\prod i{\in}P \; n. \; 1 + 1 \; / \; real \; i)$ **by** (*subst prod.reindex*) (*auto simp*: *inj-on-def*)
**also have** $\ldots \le (\prod i{\in}P \; n. \; exp \; (1 \; / \; real \; i))$ **by** (*intro prod-mono*) *auto*
**also have** $\ldots = exp \; (\sum i{\in}P \; n. \; 1 \; / \; real \; i)$ **by** (*simp add*: *exp-sum*)
**finally have** $ln \; (harm \; n) \le ln \; (\ldots * (5/3))$ **using** *n*
**by** (*subst ln-le-cancel-iff*) *simp-all*
**hence** $ln \; (harm \; n) - ln \; (5/3) \le (\sum i{\in}P \; n. \; 1 \; / \; real \; i)$
**by** (*subst* (*asm*) *ln-mult*) (*simp-all add*: *algebra-simps*)
**thus** *?thesis* **unfolding** *P-def*
**by** (*subst* (*asm*) *sum.distinct-set-conv-list*) *simp-all*
**qed**

We can use the inequality $\ln(n+1) \le H_n$ to estimate the asymptotic growth of the partial prime harmonic series. Note that $H_n \sim \ln n + \gamma$ where $\gamma$ is the Euler–Mascheroni constant (approximately 0.577), so we lose some accuracy here.

**corollary** *prime-harmonic-lower'*:
**assumes** $n$: $n \ge 2$
**shows** $(\sum p{\leftarrow}primes\text{-}upto \; n. \; 1 \; / \; real \; p) \ge ln \; (ln \; (n + 1)) - ln \; (5/3)$
**proof** −
**from** *assms ln-le-harm*[*of n*] **have** $ln \; (ln \; (real \; n + 1)) \le ln \; (harm \; n)$ **by** *simp*
**also from** *assms* **have** $\ldots - ln \; (5/3) \le (\sum p{\leftarrow}primes\text{-}upto \; n. \; 1 \; / \; real \; p)$
**by** (*rule prime-harmonic-lower*)
**finally show** *?thesis* **by** − *simp*
**qed**

**lemma** *Bseq-eventually-mono*:
**assumes** *eventually* $(\lambda n. \; norm \; (f \; n) \le norm \; (g \; n))$ *sequentially Bseq g*
**shows** *Bseq f*
**proof** −
**from** *assms*(*1*) **obtain** *N* **where** *N*: $\bigwedge n. \; n \ge N \Longrightarrow norm \; (f \; n) \le norm \; (g \; n)$
**by** (*auto simp*: *eventually-at-top-linorder*)
**from** *assms*(*2*) **obtain** *K* **where** *K*: $\bigwedge n. \; norm \; (g \; n) \le K$ **by** (*blast elim*!: *BseqE*)
**{**
**fix** *n* :: *nat*
**have** $norm \; (f \; n) \le max \; K \; (Max \; \{norm \; (f \; n) \; | n. \; n < N\})$
**apply** (*cases* $n < N$)
**apply** (*rule max.coboundedI2*, *rule Max.coboundedI*, *auto*) []
**apply** (*rule max.coboundedI1*, *force intro*: *order.trans*[*OF N K*])
**done**
**}**
**thus** *?thesis* **by** (*blast intro*: *BseqI'*)

**qed**

**lemma** *Bseq-add*:
**assumes** *Bseq* (*f* :: *nat* ⇒ *'a* :: *real-normed-vector*)
**shows** *Bseq* (*λx. f x* + *c*)
**proof** −
**from** *assms* **obtain** *K* **where** *K*: ⋀*x. norm* (*f x*) ≤ *K* **unfolding** *Bseq-def* **by** *blast*
**{**
**fix** *x* :: *nat*
**have** *norm* (*f x* + *c*) ≤ *norm* (*f x*) + *norm c* **by** (*rule norm-triangle-ineq*)
**also have** *norm* (*f x*) ≤ *K* **by** (*rule K*)
**finally have** *norm* (*f x* + *c*) ≤ *K* + *norm c* **by** *simp*
**}**
**thus** *?thesis* **by** (*rule BseqI'*)
**qed**

**lemma** *convergent-imp-Bseq*: *convergent f* ⟹ *Bseq f*
**by** (*simp add*: *Cauchy-Bseq convergent-Cauchy*)

We now use our last estimate to show that the prime harmonic series diverges. This is obvious, since it is bounded from below by $\ln(\ln(n+1))$ minus some constant, which obviously tends to infinite.

Directly using the divergence of the harmonic series would also be possible and shorten this proof a bit..

**corollary** *prime-harmonic-series-unbounded*:
¬*Bseq* (*λn.* ∑ *p*←*primes-upto n. 1* / *p*) (**is** ¬*Bseq ?f*)
**proof**
**assume** *Bseq ?f*
**hence** *Bseq* (*λn. ?f n* + *ln* (*5/3*)) **by** (*rule Bseq-add*)
**have** *Bseq* (*λn. ln* (*ln* (*n* + *1*)))
**proof** (*rule Bseq-eventually-mono*)
**from** *eventually-ge-at-top*[*of 2*::*nat*]
**show** *eventually* (*λn. norm* (*ln* (*ln* (*n* + *1*))) ≤ *norm* (*?f n* + *ln* (*5/3*))) *sequentially*
**proof** *eventually-elim*
**fix** *n* :: *nat* **assume** *n*: *n* ≥ *2*
**hence** *norm* (*ln* (*ln* (*real n* + *1*))) = *ln* (*ln* (*real n* + *1*))
**using** *ln-ln-nonneg*[*of real n* + *1*] **by** *simp*
**also have** . . . ≤ *?f n* + *ln* (*5/3*) **using** *prime-harmonic-lower'*[*OF n*]
**by** (*simp add*: *algebra-simps*)
**also have** *?f n* + *ln* (*5/3*) ≥ *0* **by** (*intro add-nonneg-nonneg sum-list-nonneg*) *simp-all*
**hence** *?f n* + *ln* (*5/3*) = *norm* (*?f n* + *ln* (*5/3*)) **by** *simp*
**finally show** *norm* (*ln* (*ln* (*n* + *1*))) ≤ *norm* (*?f n* + *ln* (*5/3*))
**by** (*simp add*: *add-ac*)
**qed**
**qed** *fact*
**then obtain** *k* **where** *k*: *k* > *0* ⋀*n. norm* (*ln* (*ln* (*real* (*n*::*nat*) + *1*))) ≤ *k*

**by** (*auto elim*!: *BseqE simp*: *add-ac*)

**define** *N* **where** *N* = *nat* ⌈*exp* (*exp k*)⌉
**have** *N-pos*: *N* > *0* **unfolding** *N-def* **by** *simp*
**have** *real N* + *1* > *exp* (*exp k*) **unfolding** *N-def* **by** *linarith*
**hence** *ln* (*real N* + *1*) > *ln* (*exp* (*exp k*)) **by** (*subst ln-less-cancel-iff*) *simp-all*
**with** *N-pos* **have** *ln* (*ln* (*real N* + *1*)) > *ln* (*exp k*) **by** (*subst ln-less-cancel-iff*) *simp-all*
**hence** *k* < *ln* (*ln* (*real N* + *1*)) **by** *simp*
**also have** ... ≤ *norm* (*ln* (*ln* (*real N* + *1*))) **by** *simp*
**finally show** *False* **using** *k*(*2*)[*of N*] **by** *simp*
**qed**

**corollary** *prime-harmonic-series-diverges*:
¬*convergent* (λ*n*. ∑*p*←*primes-upto n*. *1* / *p*)
**using** *prime-harmonic-series-unbounded convergent-imp-Bseq* **by** *blast*

**end**