

A fájl legalább Python 3.11.x-ben futtatható.

Legyen az  $n$ -edik prímszám  $p_n$  ( $p_1 = 2$ ,  $p_2 = 3$  stb.). Innen él meg  $p_{n-1} + p_n$ -be és  $p_n + p_{n+1}$ -be. Ha  $p_n$  elég nagy ( $p_n > 3$ ), akkor mindkét összeg páros, és nem prím.

Tehát a két összegből a számok prímosztóiba mennek élek. Tehát a legnagyobb szám, amivel a két összeg összeköttetésben állhat, az a  $\frac{p_{n-1}+p_n}{2}$  és a  $\frac{p_n+p_{n+1}}{2}$ . Viszont ezek két egymást követő prímszám között vannak, tehát nem lehetnek prímek, így mégsem megy beljükél. Tehát legfeljebb  $\frac{p_{n-1}+p_n}{3}$ -val és  $\frac{p_n+p_{n+1}}{3}$ -val lehetnek összeköttetésben. Viszont Bertrand posztulátuma szerint  $p_{n+1} < 2p_n$ , tehát mindkét szám kisebb  $p_n$ -nél.

Tehát ha egyszer  $p_n$ -en vagyunk, és  $p_n > 3$ , akkor onnan csak kisebb prímekre mehetünk. Ha  $p_n = 3$ , akkor onnan csak 5-re és 8-ra tudunk menni. Az előbbi esetben egy újabb prímet kaptunk, melynél nagyobb prímet már nem tudunk elérni, utóbbi esetben pedig a fenti bizonyítás teljesen jó. Ha  $p = 2$ , akkor csak 5-re tudunk menni, és ennél nagyobb prímre már nem tudunk menni.

A fentiek alapján a legnagyobb elérhető szám  $a \leq 100$  és  $b \leq 100$  esetén  $97 + 101 = 198$  (hiszen a legnagyobb, 100-nál nem nagyobb prím a 97). Úgyhogy előre kiszámoltam (kimásoltam Wikipédiáról) a 198-nál nem nagyobb prímeket. Innentől kezdve pedig egy egyszerű mélységi keresésről beszélünk, ha éppen start-on vagyunk, és már bejártuk a path-ban lévő számokat, akkor rekurzívan meghívom a keresőfüggvényt az összes olyan számra, melyre el lehet jutni start-ból, és nincsen path-ban.