

Zadanie: KOP

Kompilacja



Warsztaty ILO, grupa olimpijska. Dostępna pamięć: 128 MB.

Rozwiązanie wzorcowe $O(2^n)$

Do rozwiązania problemu użyjemy programowania dynamicznego na maskach bitowych. Zdefiniujemy tablicę $need[i]$. Zapalony j -ty bit w tej masce oznacza, że żeby skompilować i -tą bibliotekę, musi przed nią wystąpić j -ta biblioteka. Teraz zdefiniujemy maskę $mask$, każdy zapalony bit w tej masce oznacza, że ta biblioteka jest już skompilowana. Niech $dp[mask]$ przechowuje parę. Ile bibliotek już wystąpiło w naszej komendzie i jakie biblioteki już wystąpiły wcześniej (będzie to alternatywa masek $need[]$, a dokładnie tych których bity są zapalone w $mask$). Teraz musimy wiedzieć jak do $mask$ dokompilować kolejną bibliotekę i . Musimy sprawdzić ile bibliotek musimy dopisać do aktualnej komendy. skoro $dp[mask].second$ mówi nam jak wygląda aktualna komenda, to wystarczy sprawdzić ile jest takich miejsc, gdzie jest zero w masce $dp[mask].second$ i jedynka w $need[i]$. Powie nam to ile bibliotek musimy dokompilować. Teraz wystarczy zaktualizować nasze wartości w $mask \vee 2^i$. Ogólna złożoność rozwiązania, to $O(2^n)$.