

Úloha č. 1

Silueta města

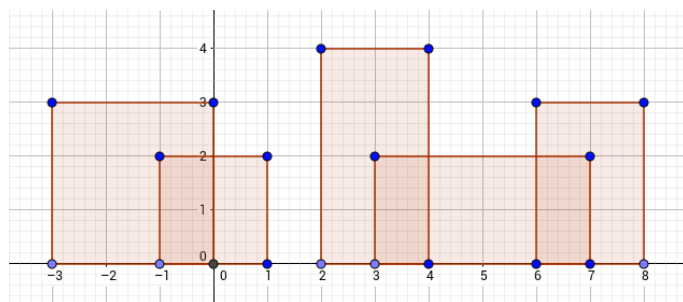


Odpověz Sfinze!

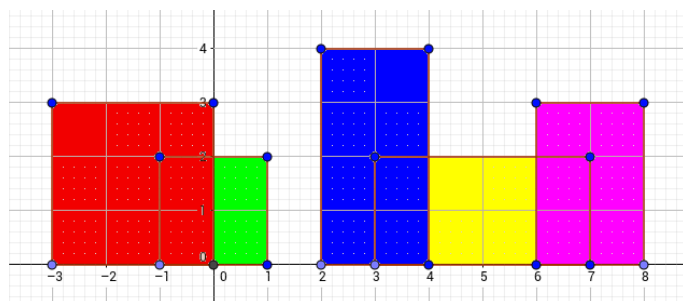
10 b

Tato úloha je vyhodnocována automaticky. Je potřeba, aby výstup programu **přesně** korespondoval se specifikací výstupu níže. Jak odevzdávat tento typ úloh se můžeš dočíst na webových stránkách FIKSu pod záložkou „Jak řešit FIKS“.

Naším cílem je spočítat plochu, kterou zabírá silueta města. Povšimneme si, že plochu můžeme sestavit z nepřekrývajících se obdélníků, které leží na x -ové ose a mohou mít různou výšku a šířku. Dále pozorujeme, že obdélníky mohou začínat či končit pouze tam, kde nějaká budova končí či začíná. Například pro třetí ukázkový vstup (viz obr. 1.1) můžeme nasčítávat plochy těchto obdélníků jako na obr. 1.2.



Obrázek 1.1 Nákres třetího vstupu



Obrázek 1.2 Rozdělení třetího vstupu na obdélníky

Postupně tedy zleva doprava procházíme zajímavá místa, tedy body, ve kterých nějaká budova začíná či končí. Pokud se stane, že se v daném bodě změní výška siluety města, připočteme k obsahu nalezený obdélník, který se nachází nalevo od daného bodu. Plochu spočteme jednoduše jako součin dosavadní výšky siluety města a vzdálenosti k poslednímu minulému započítávanému obdélníku. Obsah může být samozřejmě i nulový, například pokud je mezi budovami mezera.

Složitost algoritmu

Vzhledem k časové složitosti potřebujeme projít všechny body, kde nějaká budova začíná nebo končí. Těch je $O(N)$, kde N je počet budov. Nejprve si potřebujeme tyto body seřadit zleva doprava. Řazení nám zabere $O(N * \log(N))$ času. Pro účinné vyhledávání největší výšky budovy na intervalu je vhodné použít strukturu, která je schopná hledat maximum z momentálně se překrývajících budov v rychlejším čase než lineárním. Tuto hodnotu je třeba i často aktualizovat. Nesmíme také zapomenout, že překrývající se budovy mohou mít stejnou výšku. Např. v jazyku C++ můžeme použít multiset. Časová složitost zjištění momentálně nejvyšší budovy na daném intervalu je tedy $O(\log(N))$. Protože aktuálně nejvyšší budovu musíme zjišťovat v každém bodě, kde končí nebo začíná budova, celková časová složitost je $O(N * \log(N))$.

Pokud posuzujeme paměťovou složitost, je nutné si uchovávat všechny body, kde budovy začínají či končí. Paměťová složitost je tedy $O(N)$.

V lehčích variantách bylo také možné řešit úlohu pomocí "obarvování" dvourozměrného pole. Tento postup však nebylo možné použít na těžší verzi, protože jeho časová i prostorová složitost je úměrná ploše, na které se budovy mohou nacházet.