

Řešení úlohy č. 3

Ryby

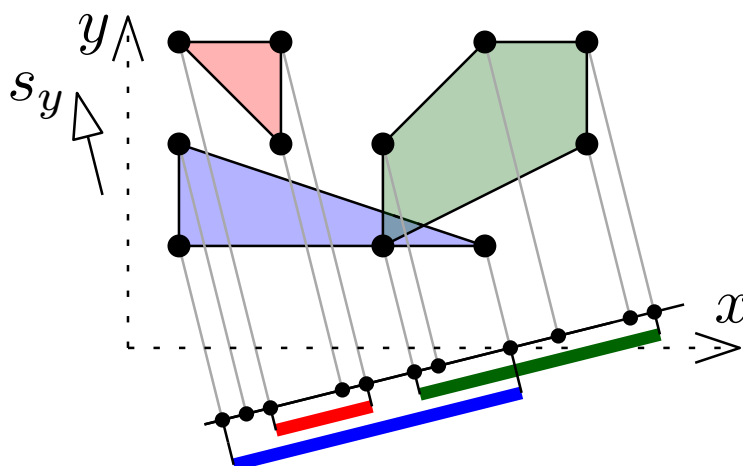
Formálně problém Ryby můžeme představit takto: Vstupem do programu je seznam N konvexních polygonů a směrový vektor (s_x, s_y) . Určete přímku takovou, že protíná maximální počet zadaných konvexních polygonů.

Protože známe směrový vektor, tak můžeme zobrazit (projekcí) všechny konvexní polygony na přímku, která je kolmá ke směrovému vektoru. Toto lze počítat buď pomocí metod výpočetní geometrie, která ovšem do výpočtů může zanést nepřesnosti (pokud použijeme desetinná čísla).

Nepotřebujeme přesně onu projekci, ale pouze pořadí již projektovaných bodů. Pořadí bodů u a v po projekci lze získat analyticky pomocí vektorového součinu

$$w = v - u \quad r = s_x \cdot w_y - s_y \cdot w_x.$$

Pokud výsledek těchto operací r je pozitivní, tak v je v pořadí dříve než u , pokud je negativní, tak v je v pořadí později než u , a pokud je 0, tak jsou zobrazeny na stejné místo.



Ilustrace: Konverze tvarů na intervaly

Povšimneme si, že z každého konvexního útvaru potřebujeme v pořadí pouze první a poslední bod, protože všechny přímky, které začínají mezi nimi budou útvar protínat.

Abychom zjistili maximální počet útvarů, které přímka může protnout, budeme body zpracovávat postupně. Začneme s čítačem na hodnotě 0. Pokaždé, když narazíme na první bod skupiny, tak zvýšíme čítač o jedna. Pokaždé, když narazíme na poslední bod skupiny, tak snížíme čítač o jedna. Pamatujeme si maximum, kterého čítač během tohoto procesu dosáhl.

Protože chceme započítat útvary i v případě, kdy se ho přímka i pouze dotýká, tak potřebujeme zpracovat body, které jsou zobrazeny na stejné místo v projekci, v pořadí tak, aby se započítaly všechny útvary. Proto chceme nejdříve zpracovat počáteční body konvexních útvarů (tím je přidat), a až potom přidat koncové body konvexních útvarů (což je odebrat). Tento požadavek lze zpracovat do řazení: pokud je $r = 0$, tak vrchol, který je počáteční bude v pořadí dříve. (Kdybychom nechtěli okraje započítávat, tak bychom při $r = 0$ řadili opačně.)

Analýza složitosti

Popsaný postup vyžaduje mimo vstupu i paměť na uložení:

- $6 \cdot N = O(N)$ prvního a posledního bodu pro každý útvar společně se značkou, jestli je bod první či poslední v pořadí.
- Konstantní počet $= O(1)$ pro pracovní proměnné.

Popsaný postup má časovou složitost:

- Operace pro upravené porovnávání dvou prvků používá zhruba 10 instrukcí, což je $O(1)$.
- Na zjištění maxima a minima u každého útvaru trvá lineární čas (dvě porovnání pro každý prvek a podmíněné přiřazení), což je $O(M)$ (kde M je počet prvků útvaru).
- Standardní algoritmus řazení K prvků používá $O(K \log K)$ porovnání (a dalších instrukcí). Protože řadíme $2N$ prvků, a jedno porovnání trvá $O(1)$, dostáváme $O(2N \log 2N) = O(N \log N)$.

Označme velikost vstupu jako R . (Zadání specifikuje pouze počet tvarů. Každý tvar může mít nanejvýš $8 \cdot 10^3$ bodů, ale definování R nám umožňuje zachytit časovou složitost přesněji.) Celková paměťová složitost je $O(N)$ a celková časová složitost je $O(R + N \log N)$.