Páska

Pro tento úkol jsem použil Chanův algoritmus.

Všechny hodnoty na vstupu nejprve postupně uložím do neuspořádané množiny unikátních bodů, abych předešel možné duplikaci bodů na vstupu a uložil jen unikátní body. Po uložení bodů je rozdělím na k skupin po m bodech (v první iteraci je m = 1<<1<<t, t = 0, => m = 2). Pro každou skupinu poté vypočítám obal pomocí monotone chain algoritmu:

Body seřadím od nejmenší souřadnice x po největší. Nejprve vytvořím jednu stranu obalu;

Dokud přímka: [předposlední bod v obalu -> poslední bod v obalu -> nový bod] “zatáčí” doprava nebo rovně, tak odeberu poslední bod v obalu a přidám do něj ten nový bod. Pokud přímka zatáčí doleva tak nový bod zahodím. Tento postup opakuji pro všechny body. Zároveň vytvářím I obal z druhé strany tím, že body používám v otočeném pořadí. Jednoduše poté obě strany obalu spojím dohromady.

Po vytvoření všech pod-obalů pracuji se dvěma body pro vytvoření celkového obalu: Poslední bod ve finálním obalu (na začátku to je bod nejvíce vlevo dole) a nadcházející bod ze stejného pod-obalu. K těmto dvěma bodům poté postupně přiřadím zvolený nejoptimálnější bod z každého pod-obalu a zvolím ten, se kterým přímka těchto tří bodů zatáčí nejvíce doprava. Pokud žádná z přímek nezatáčí doprava, dalším bodem v obalu se stává právě ten nadcházející bod ze stejného pod-obalu.

Zvolení optimálního bodu v pod-obalu:

Protože jsou body v pod-obalech seřazeny, vyberu optimální bod pomocí binárního vyhledávání, kde optimálním bodem je bod, jehož oba sousedé v přímce: [předposlední bod ve finálním obalu –> poslední bod ve finálním obalu –> soused] oba zatáčejí doleva.

Tento postup vytváření finálního obalu opakuji m-krát, pokud se po m iteracích obal nedokončí, zvětším počet bodů m (m = 1<<1<<t, t+=1, => m = 4). Poté celý process opakuji odznova dokud m >= h, kde h je počet bodů ve finálním obalu.

Obal bude vypočítán v čase O(n\*log(h)), kde n je celkový počet bodů a h je počet bodů v obalu.