

Konveksna ovojnica v ravnini

Lekcija 4

Sergio Cabello

`sergio.cabello@fmf.uni-lj.si`

FMF

Univerza v Ljubljani

Zavijanje darila

Algorithm *Gift-wrapping*

Input: Množica točk P

Output: Seznam oglišč za $CH(P)$

1. Najdemo najnižjo točko q_0 od P ;
2. $i \leftarrow 0$;
3. **repeat**
4. $i \leftarrow i + 1$;
5. q_i je taka točka, da je q_{i-1}, q_i, q zasuk na desno za vsak $q \in P$;
6. **until** $q_i = q_0$
7. **return** q_0, q_1, \dots, q_{i-1} ;

Zavijanje darila – Splošni položaj

Kje imamo lahko probleme?

Veliko točk je lahko kandidatinj ali pa so na premici.

Algorithm *Gift-wrapping*

1. Najdemo **najnižjo točko** q_0 od P ;
2. $i \leftarrow 0$;
3. **repeat**
4. $i \leftarrow i + 1$;
5. q_i je taka točka, da je **q_{i-1}, q_i, q** zasuk na desno za vsak $q \in P$;
6. **until** $q_i = q_0$
7. **return** q_0, q_1, \dots, q_{i-1} ;

Gramahov algoritem - Prva opcija

Izgradimo M in ga spreminjamo. Potrebujemo kazalce.

Algorithm *Grahamov algoritem*

1. Najdemo najnižjo točko $p_0 = p_*$;
2. $(p_1, \dots, p_{n-1}) \leftarrow$ uredimo točke $P \setminus \{p_*\}$ po kotih od p_* ;
3. Naredimo krožni seznam $L = (p_0, p_1, \dots, p_{n-1})$
4. $p \leftarrow p_0$; $q \leftarrow p_1$; $r \leftarrow p_2$;
5. **while** $q \neq p_0$
6. **if** p, q, r zasuk na desno
7. $p \leftarrow q$; $q \leftarrow r$; $r \leftarrow r.next$;
8. **else** brišemo element q od L ;
9. $q \leftarrow p$; $p \leftarrow p.prev$;
10. **return** seznam L minus zadnja p_0 ;

Ali je vrstica 5 problematična, če je p_0, p_1, p_2 zasuk na levo?

Gramahov algoritem - Druga opcija

Ne izgradimo M na začetku, temveč uporabimo za delo sklad M . Naredimo ga lahko s tabelo.

Algorithm *Grahamov algoritem*

1. Poiščemo najnižjo točko p_* ;
2. $(p_1, \dots, p_{n-1}) \leftarrow$ urejamo točke $P \setminus \{p_*\}$ po kotih od p_* ;
3. S prazen sklad; damo p_* in p_1 na sklad;
4. $i \leftarrow 2$, **while** $i < n$
 5. $q \leftarrow$ vrh S ; $p \leftarrow$ element pod vrhom S ;
 6. **if** p, q, p_i zasuk na desno
 7. damo p_i na sklad S ;
 8. $i \leftarrow i + 1$;
 9. **else** odstranimo element q s sklada S ;
10. **return** elementi sklada S