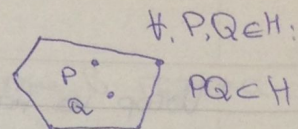
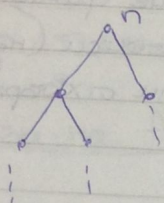


24 октомври 2013г. - ИГ - лекция



$\sim n \cdot \lg n \rightarrow$ оптимална скорост на подреждане



алгоритми за подреждане
 $\hookrightarrow$ обр. двоични дървета

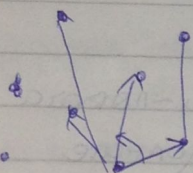
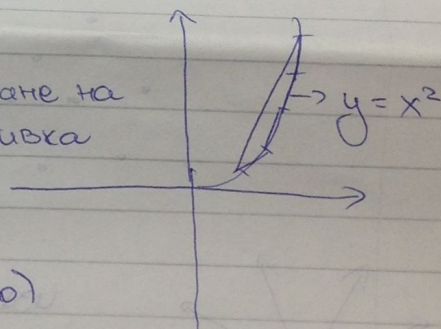
дървото има поне $n!$ листа
 $\log_2 n! \rightarrow$ поне височина

Гурилинг $\log n! \approx n \lg n$

Имаме да подредим числата $\{x_i\}_{i=1}^n$ (x_i, x_i^2)

$x_i \geq 0$

$\leftrightarrow$ намиране на обвивка



Метод на Jarvis
 (на опаковането)

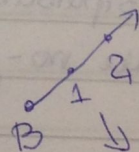
1. Взимаме точката с $\min y$ координата (не е задължително)
2. Избираме най-дясната от останалите (предимство на вектори)

h - върховете от обвивката

n - всички върхове

$h \cdot n \rightarrow$ сложност

worst-case регр. n^2 , но работи добре на добре конфигурирани данни

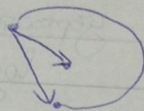


пренебрегване
 по близостта
 от двете
 точки

Избор на начална точка $\rightarrow$ от всички най-долни

Край на алгоритъма

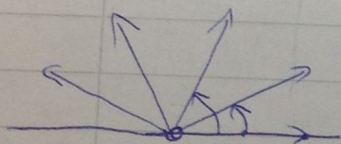
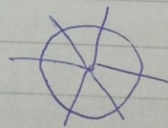
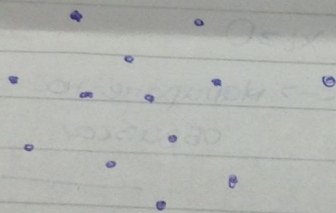
Пр. $\rightarrow$ оставяме началната точка сред другите и когато тя се окаже избрана отново $\rightarrow$ край [повече сравнения]
Избиране специални точки докато имаме точки (начална точка, защото точките са или избрани или отхвърлени)



Алгоритъм на Graham

оптимална скорост

използване алгоритъм за подреждане

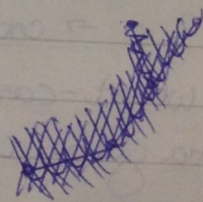
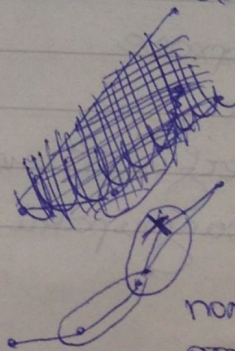


Избор на 1^{ва} точка най-лявата от най-долните

Библиотечен алгоритъм за подреждане
 $\hookrightarrow$ сортир. с предпоставянето

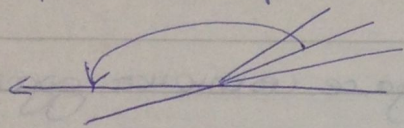
Изтриване по-близките точки ако на общия вектор има по-далечна точка

Обход. е линеен процес за всяка точка
 $\hookrightarrow$ добавяне в стека еднократно
 $\hookrightarrow$ може да се наложат да 2 изтриени макс еднократно



положителна vs. отрицателна тройка $-2-$

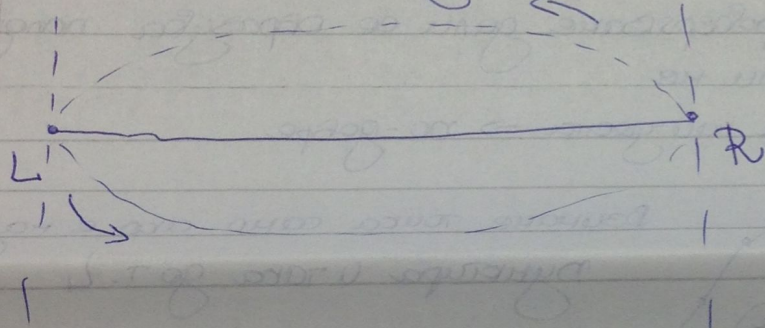
Край на алгоритма $\rightarrow$ точката от последния път в горната полуравнина



$\Theta(n \log n) \Rightarrow$ оптимален алгоритъм

Алгоритъм на Andrew

от ляво-отвд от точките да опр. най-лявата и най-дясната

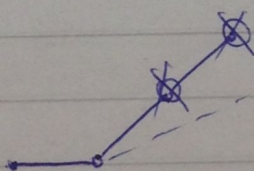


свс сегурност L и R са от обвивката

Разгл. горните и долните точки поотделно

Подрязваме точките по x
 $x_i < x_{i+1}$

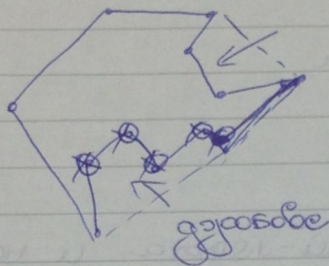
Групираме точките, както при Грехам (полюс. vs. стр. кар. тройка)
 изтр. е окончателно



изтриване докато
 има отриц. ориент. тройка

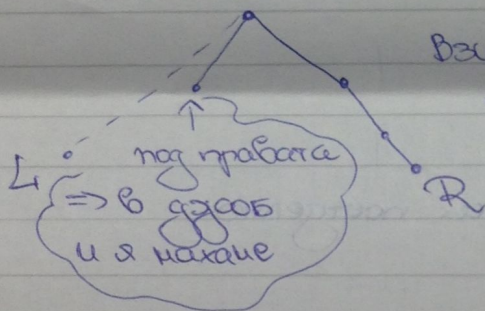
Алгоритъм, който намира източната обвивка на прост многоъгълник

Целта е да се източат дървовите



Линейна скорост, защото върховете
опазват посока поперек

Трупаме точки и проверяваме дали се образува поперек.
ориентирана тройка или не
ако разгн. най-ляво и най-дясно $\rightarrow$ по-добре



Взимаме точка само ако е над
пунктира и така до т. L