Scientific Computing

Feb 10, 2025

Announcements

→ HW 2 due Mon. Feb. 17

→ Office hours this Friday are rescheduled to 2pm-3pm.

→ Next Monday, no in-person lecture and no office hours.

Today

→ Homework 1 Solutions

→ Search Spaces and Brute Force

Office Hours:

Mon + Fri

9:30am-10:30am

Cudahy 307

The search space of a problem is the set of all possible "things" that may or may not satify your constraints and that all have some score that you want to minimize or maximize.

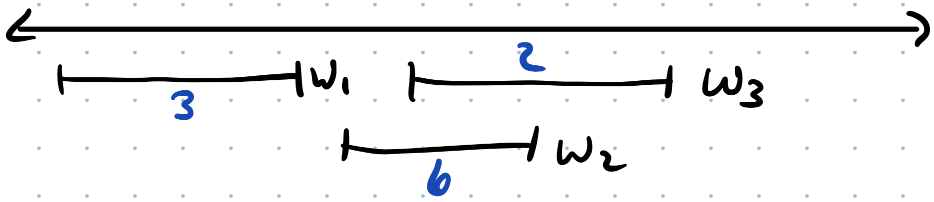
The next few lectures are focused on ways to actually check the entire search space to find the optimal solution.

^
guaranteed

The most obvious way to do this is brute force: generate every single element of the search space, and check whether it satisfies the constraints and if so, what its score is.

Ex 1: Weighted Interval Scheduling

3 requests



Search space: all subsets of $\{w_1, w_2, w_3\}$

Candidate	satisfies constraints	score
$\{\}$	✓	0
$\{w_1\}$	✓	3
$\{w_2\}$	✓	6
$\{w_3\}$	✓	2
$\{w_1, w_2\}$	✓	9 optimal
$\{w_1, w_3\}$	✓	5
$\{w_2, w_3\}$	X	8 – irrelevant
$\{w_1, w_2, w_3\}$	X	11 – irrelevant

Fact: There are 2^n subsets of a set of size n.

$n = \#$ of Requests

looping over each thing once

Pseudocode

```
R = set of requests
b = 0
for each subset r of R:
    if r is valid:
        s = score(r)
        if s > b:
            b = s
return b
```

$O(1)$

loops 2^n times

validity check takes $O(n)$

scoring takes $O(n)$

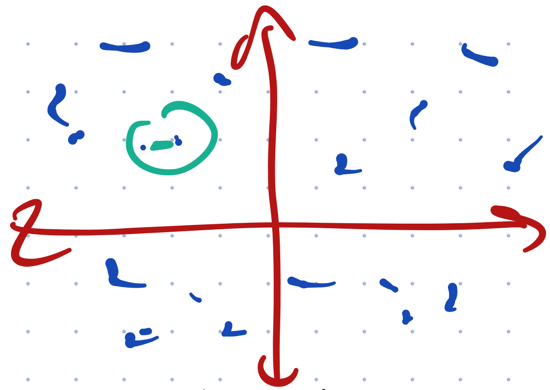
total time $\sim 2^n \cdot \underbrace{(2n)}_{n+n} = O(n 2^n)$

$O(2 \cdot n \cdot 2^n)$

Knapsack: Same situation. n items

Search space = all subsets of those n items

size is 2^n again



<u>Closest Pair</u>: Input n points in the xy-plane

Goal: Find the closest pair (normal Euclidean distance).

Search space = all <u>unordered</u> pairs of points

$(p_1, p_2) = (p_2, p_1)$

Suppose our points are $\{p_1, p_2, p_3, p_4\}$.

The search space is: $\{(p_1,p_2), (p_1,p_3), (p_1,p_4), (p_2,p_3)$
$(p_2,p_4), (p_3,p_4)\}$ 6 pairs

nC_2

In general, it's the binomial coefficient $\binom{n}{2}$ "n choose 2" which is the # of ways of picking 2 things out of n.

$$\binom{n}{2} = \frac{n(n-1)}{2} = \frac{1}{2}n^2 - \frac{1}{2}n = O(n^2).$$

ignores multiples
and vanishing terms

Quadratic, not Exponential

* It may seem surprising, but this can actually be done in $\underline{O(n \cdot \log(n))}$ time, so without checking every pair! (next lecture)

slightly slower than $O(n)$
much faster than $O(n^2)$