Brute Force

P : problem to solve.

c : solution candidate

```
c ⇐ first(P)
while  c ≠ Λ {
    if valid(P, c) {
       output(P, c)
    }
    c ⇐ next(P, c)
}
```

first(P) : "first" candidate solution (to P)

next(P, c) : "next" candidate solution "after" c

valid(P, c) : is c a solution to P?

output(P, c) : "save" c

Λ : no more solution candidates

Example: Selection sort.

(1, 1 : one
7 : seven)

```
sml := 1;
For i from 1 to n {
    for j from i to n {
        if A[j] < A[sml] {
            sml := j;
        }
    }
    tmp := A[i];
    A[i] := A[sml];
    A[sml] := A[tmp];
}
```

(rough) operation count (worst case)

1

5

5, 7, 12, loop, $12(n-i+1)$

5, 2

3, 4, 4, 11

$12(n-i+1)+11$

$12(n-i+1)+16$

$$1+\left(\sum_{i=1}^{n} 12(n-i+1)+16\right)$$

– cont. page 2

$$\cancel{= 1+\sum_{i=1}^{n} 12(n-i+1)+\sum_{i=1}^{n}16}$$

$$\cancel{= 1+12\sum_{i=1}^{n}(n-i+1)+16n}$$

$$=$$

Brute force how/why:

First(P) = A

next(P, c) = "one run of outer loop"

valid(P, c) = true only when A is sorted

output(P, c) = no-op

Λ : outer loop done

Operation counting

- How: See 1.
- Massaging result, example:

$$\underbrace{1 + \left(\sum_{i=1}^{n} 12(n-i+1) + 16 \right)}_{+} = 1 + \sum_{i=1}^{n} 12(n-i+1) + \sum_{i=1}^{n} 16$$

$$= 1 + 12\sum_{i=1}^{n} (n-i+1) + 16n$$

$$= 1 + 12\left(\sum_{i=1}^{n} n - \sum_{i=1}^{n} i + \sum_{i=1}^{n} 1 \right) + 16n$$

$$= 1 + 12\left(n^2 \underbrace{- \left(\tfrac{1}{2}(n^2+n)\right) + n}_{*} \right) + 16n$$

$$= 1 + 12\left(\tfrac{1}{2}(n^2+n) \right) + 16n$$

$$= 1 + 6n^2 + 6n + 16n$$

$$= 6n^2 + 22n + 1$$

- Operations performed by algorithm, (roughly, worst-case,) for A of length n:

$$f_w(n) = 6n^2 + 22n + 1$$

- best case: "if" always false.
  sml:=j never executed.
  replace $12(n-i+1)$ w. $10(n-i+1)$ in $+$,
  massage... yields

$$f_b(n) = 5n^2 + 21n + 1$$

Asymptotics

We use O, Ω and Θ to classify algorithms according to efficiency.

Def: f is in $O(g)$
if
for some $k \geq 0$
for some n_0,
for all $n \geq n_0$,
$f(n) \leq kg(n)$. ◇

asymptotic upper bound

Def: f is in $\Omega(g)$
if
for some $k > 0$
for some n_0
for all $n \geq n_0$
$f(n) \geq kg(n)$ ◇

asymptotic lower bound

(assuming $f(n)$, $g(n)$ positive)

Def f is in $\Theta(g)$
if
f is in $O(g)$
and
f is in $O(g)$ ◇

asymptotically tight bound

$O(g)$ is a set of functions which "grow" at most "as fast" as g.

$\Omega(g)$ ——— " ——— at least "as fast" as g.

$\Theta(g)$ ——— " ——— like g

How fast (worst case) running time grows as n increases is a good efficiency indicator.

Asymptotics, examples:

$f := 100$. $g := 1$.

f is in $O(g)$.

proof: pick $k = 100$,
$n_0 = 0$.
for all $n \geq 0$, $100 \leq 100 \cdot 1$ $\square$

a Full proof needs to prove this also.

One idea: use derivatives. This is almost always obvious in our case of time complexity analysis

So we never ask you to do this.

f is in $\Omega(g)$.

proof: pick $k = 1$,
$n_0 = 0$,
for all $n \geq 0$, $100 \geq 1$. $\square$ "obvious".

f is in $\Theta(g)$.

proof: f is in $O(g)$
and $\Omega(g)$. by def. of Θ.
we are done. $\square$

$f := 100, \quad g := n.$

f is in $\mathcal{O}(g)$.

proof: $k = 1$

$n_0 = 100$

$\left.\begin{array}{l}\text{for all } n \geq 100 \\ 100 \leq n.\end{array}\right\}$ "obvious" $\blacksquare$

$f := n^2, \quad g := n.$

f not in $\mathcal{O}(g)$.

proof: "invert" definition of $\mathcal{O}$.

To show: for all k, n_0

there is an $\hat{n} \geq n_0$, s.t.

$F(n) > k\,g(n).$

$\hat{n} = \max(n_0, k) + 1.$

$F(\hat{n}) = \hat{n} \cdot \hat{n} > k \cdot \hat{n} = g(\hat{n}).$ $\blacksquare$

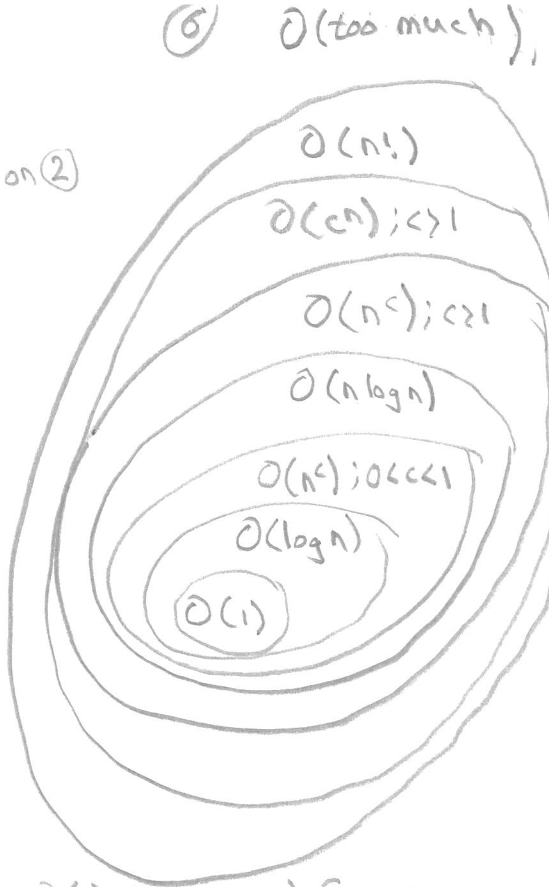
you can be slightly more sloppy than I was in example on 2

Time complexity analysis

worst case:

1. find f_w
2. find smallest $O(g)$ in …
3. such that $f_w \in O(g)$.
4. try to show $f_w \in \Omega(g)$.

(can't: maybe 2. wasn't good enough)



$O(1) \subseteq O(\log n) \subseteq \dots$

best case:

1. find f_b
2. Find smallest $\Omega(g)$ in …
3. such that $f_b \in \Omega(g)$.
4. try to show $f_b \in O(g)$.

"inverted"

$\Omega(1) \supseteq \Omega(\log n) \supseteq \dots$

Example, from 2

$f_w = 6n^2 + 22n + 1$
$g = n^2$

f_w is in $O(g)$. — a

proof. pick k s.t. $kn^2 \geq 6n^2 + 22n + 1$
candidate: $k = 29$.
for all $n \geq n_0 = 1$,

$$f_w(n) = \underbrace{6n^2 + 22\underline{n} + \underline{1} \leq 6n^2 + 22\underline{n^2} + 1\underline{n^2}}_{\text{"obvious}} = 29n^2 = kg(n)$$

$\boxed{}$

f_w is in $\Omega(g)$. — b

proof. pick k s.t. $6n^2 + 22n + 1 \leq kn^2$
candidate: $k = 6$.
for all $n \geq n_0 = 1$,

$$f_w(n) = 6n^2 + \underbrace{22n + 1 \geq 0}_{\text{"obvious"}} + 6n^2 = kg(n)$$

$\boxed{}$

f_w is in $\Theta(g)$.

proof. follows from a & b.

$\boxed{}$

Stable Matching

M, W ; $|M| = |W| = n$ men, women

$P_M : M \times W \to \mathbb{N}$

$P_W : W \times M \to \mathbb{N}$

preference } bijection on $\{1, \dots, n\}$

matching : $S \subseteq \hat{S}$ for some $\hat{S} : \underbrace{M \to W}_{\text{bijection}}$

perfect matching : $S : M \to W$

instability : for some m, w, m', w',

$S(m) = w$ $P_M(m, w') > P_M(m, w)$

$S(m') = w'$ $P_W(w', m) > P_W(w', m')$

(then m and w' would be happier if they drop their partner & marry.)

stable : perfect
& no instability.

Algorithm for computing a stable S :

Gale-Shapley. page 21.

Correctness proof included there.

Solved Exercise 2

$F \subseteq M \times W$ disallowed marriages.

instability: 4 criteria

i) usual def.

ii) $S(m) = w$, $w' \notin im(S)$, $P_M(m, w') > P_M(m, w)$, $(m, w') \notin F$

iii) — " —, $m' \notin dom(S)$, $P_W(w, m') > P_W(w, m)$, $(m', w) \notin F$

iv) $m \notin dom(S)$, $w \notin im(S)$, $(m, w) \notin F$.

Solution:

Algorithm: add

... for which $(m, w) \notin F$

at end of while criteria in G-S.

New algorithm: Yields stable matching.

proof: To show: None of i) - iv) happen.

assume (towards contradiction) i):

$\Rightarrow$ m proposed to w' (else contradiction of $P_M(m, w') > P_M(m, w)$)

w' rejected m for an already engaged partner (by algorithm)

$\Rightarrow$ m, w' cannot both prefer each other; contradiction.

assume iii):

$\Rightarrow$ m' proposed to w (else ... see above)

w rejected m for an already engaged partner (by alg)

$\Rightarrow$ w prefers current partner; contradiction.

assume ii):

$\Rightarrow$ no one (not even m) proposed to w'

$\Rightarrow$ m cannot prefer w' to current partner; contra.

assume iv):

$\Rightarrow$ m single implies m proposed to all w where $(m, w) \notin F$.

So either w rejected m for another; no longer single or $(m, w) \in F$; contradiction. ∎