K^{th} element

min — $O(n)$ $K=0$

max — $O(n)$ $K=n-1$

~~K^{th}~~ median — $O(n \log n)$ sort, ~~return~~ return $a[\frac{n}{2}]$

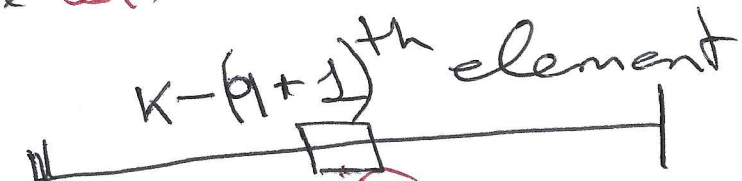
$K=\frac{n}{2}$

Idea: partition. K^{th}

case $K<q$

remove

case $K>q$



case $q=K$

solution

```
KTM ( a, k, n )
// if (n==1); k=0 , return a[0]; ?
   do not need line "else" has base init
   q = partition( a, n );
   if( k<q) KTM ( a, k, q ); // left
   elsif( k>q) KTM (a+q+1, k-q-1, n-q-1);
   else return a[k] ; // k=q
```

Side note (chat)

median-pivot QS

guess

pivot = median (a[i], a[j], a[k])

worst $T(n) = T(2) + T(n-2) + n$

$O(n^2)$

arith. $\rightarrow n^2$

geom. $\rightarrow n \log n$

for any fixed d:

a) $T(n) = T(n-d) + T(d) + n \rightarrow O(n^2)$

b) $T(n) = T(n/d) + T\left(\frac{d-1}{d}n\right) + n \rightarrow O(n \log n)$

---

median of 3 (a)

5 (a)

7 (a)

---

n (b) $d=2$

$\rightarrow \frac{n}{2}$ (b) $d=4$

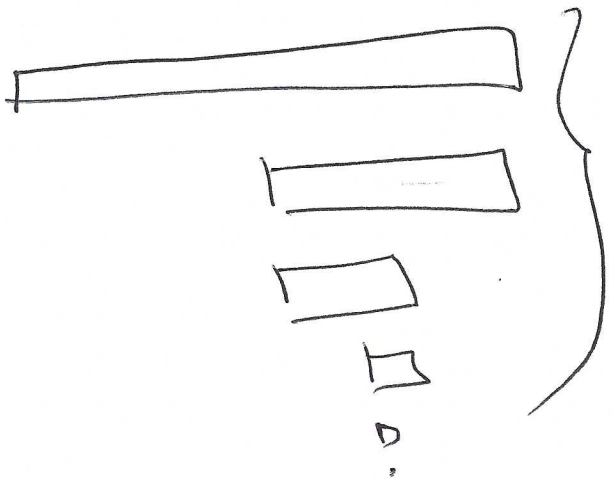
RTof K^{th}

$T(n) = \cancel{T(1)} + T(n-1) + n$

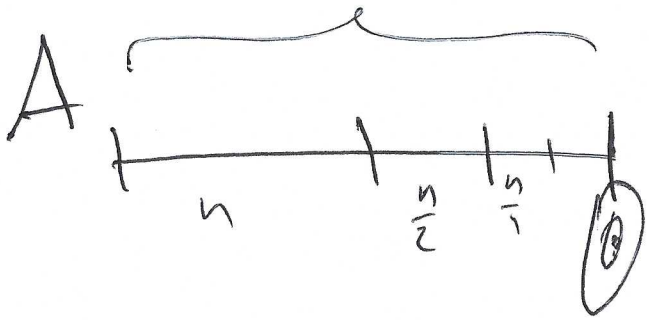
$T(n) = T(\frac{n}{2}) \cancel{+ T(\frac{n}{2})} + n$

worst $O(n^2)$

~~best~~ average... $O(n)$



n
$+ n/2$
$+ n/4$
$\vdots$
$+ 4$
$+ 2$
$+ 1 = 2n-1$



$1 + 2 + \ldots + 2^{\log_2 n} =$

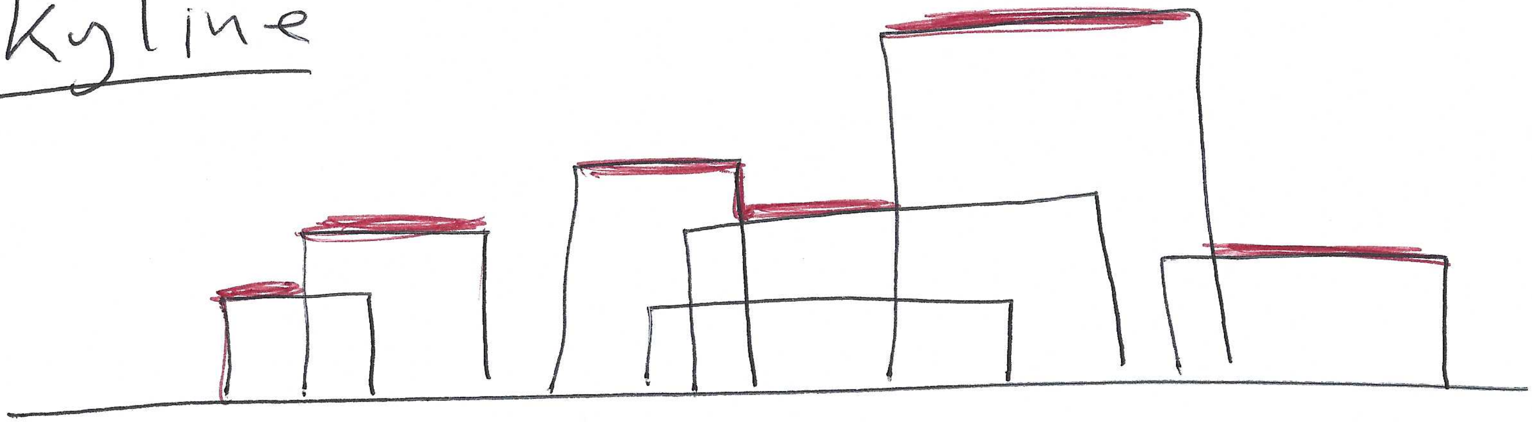
$= \frac{2^{\log_2 n + 1} - 1}{2-1} =$

$= \frac{2n-1}{1} = 2n-1$

worst case linear

median

Skyline



D&C

$T(n) = 2T(\frac{n}{2}) + n$ ← 2 ptr. tr

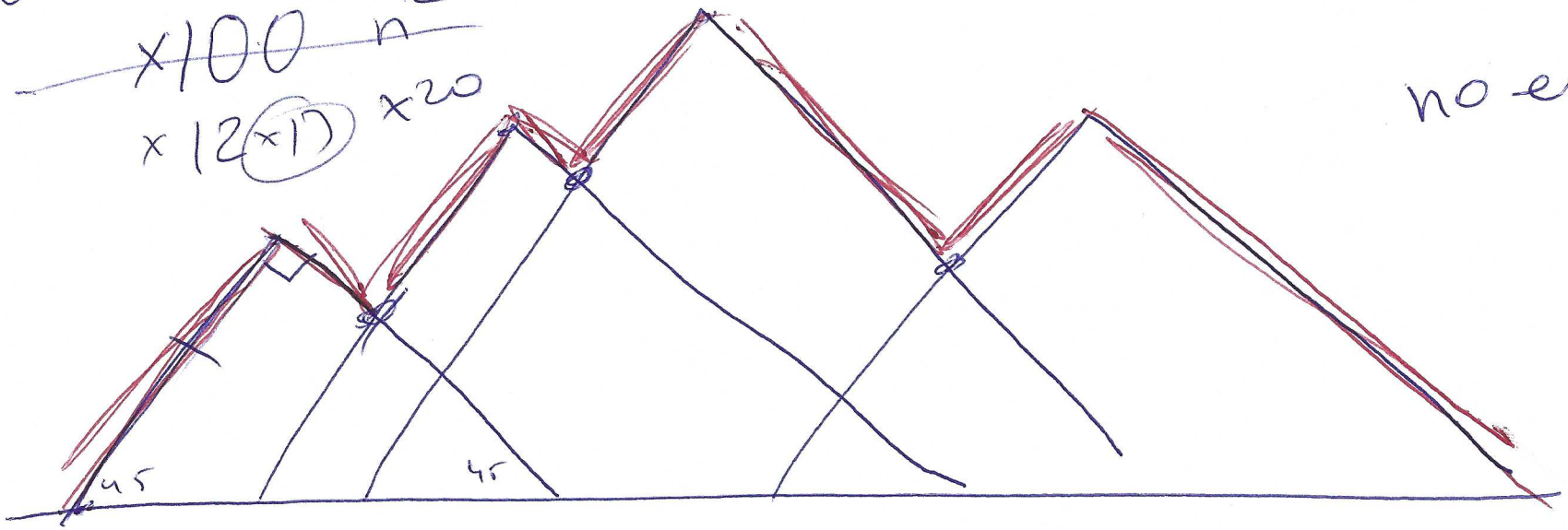
$O(n \log n)$

1000 → 10 000

~~×100~~ n^2

×12 (×17) ×20

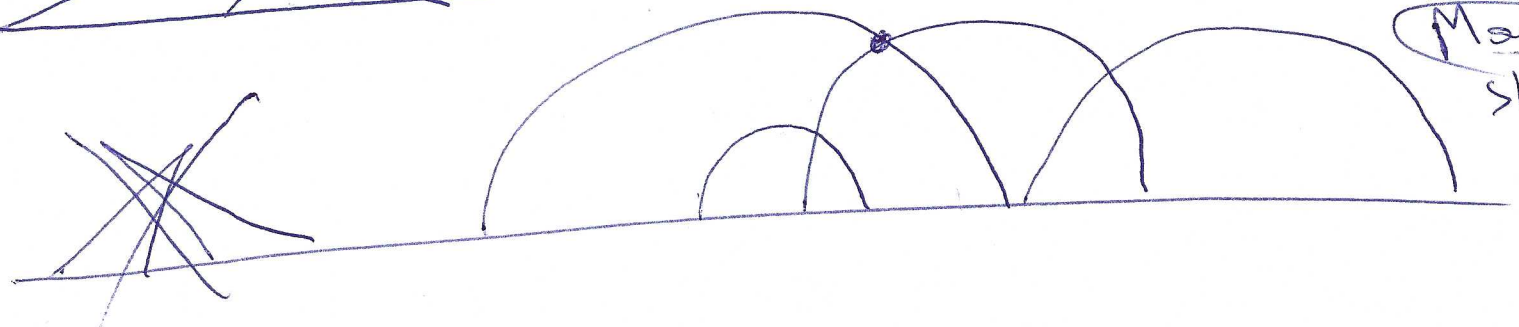
no extra



isosceles

~~+250%~~ extra

+30% if code nice



Mars skyline

Greedy = sequence

1) locally optimal ←
2) irreversible ←
steps

globally optimal = best globally

Coin change:

X 25, 10, 5, 1

↓

X = 47 = 25 + 10 + 10 + 1 + 1 + 1 + 1

7

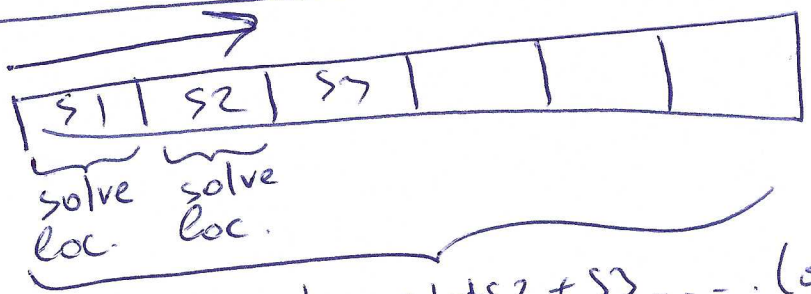
X 10, 7, 1 ↓ subopt.

↓

X = 14 → 10 + 1 + 1 + 1 + 1

→ 7 + 7 !!!

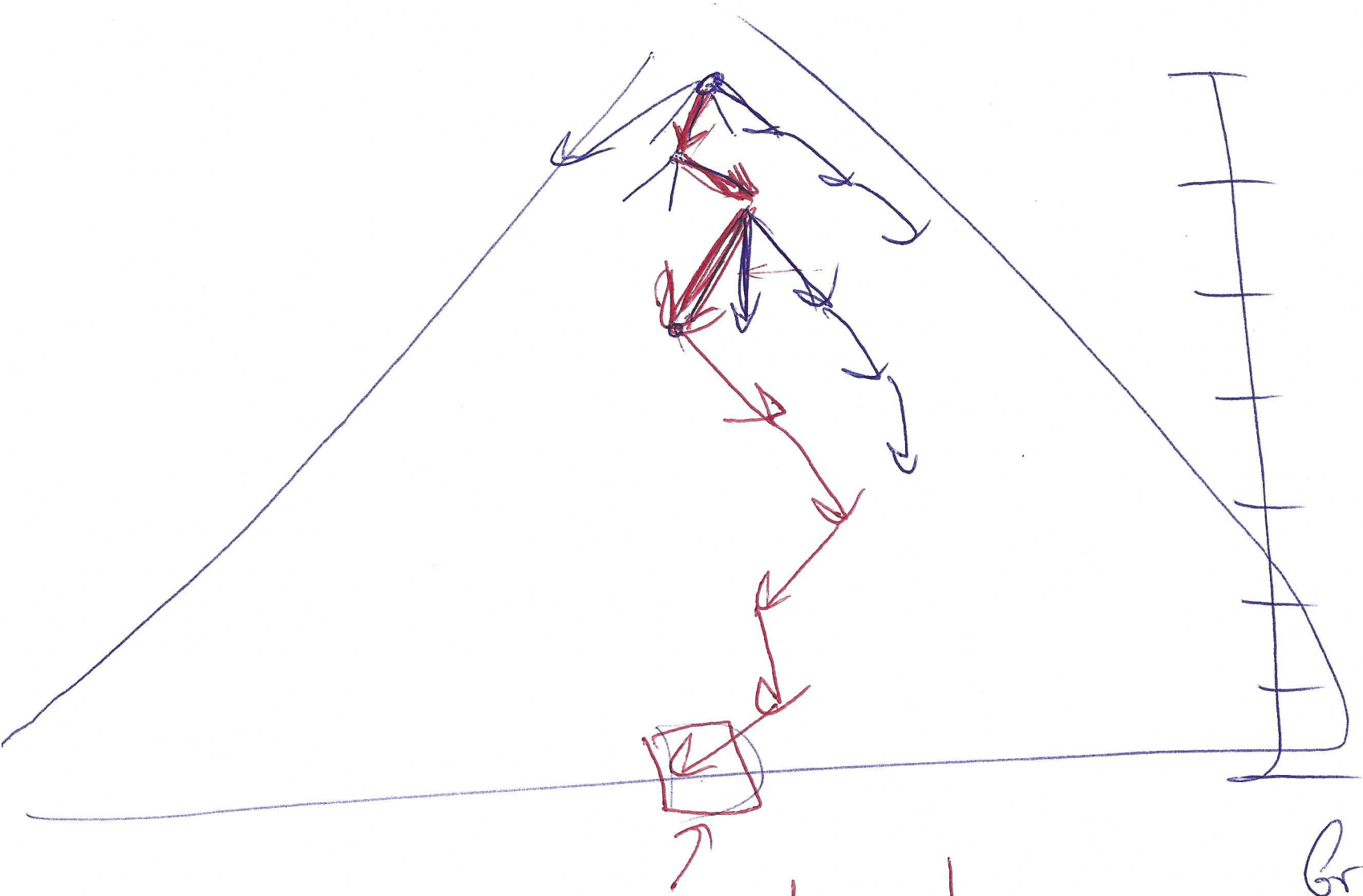
glob. opt



sol = S1 + S2 + S3 .... (greedy)

sol = ~~combine~~ (S1, S2, ...) (DAC)

1) fast ←
2) easy to create ←
3) often do not work



greedy sol.

Brute force

if irrev. is dropped you get BF

$RT(BF) =$ size of tree $= B^h$

$RT(G) =$ height $= h$