Master theorem

Assume $n = b^k$

$$\frac{n}{b} = b^{k-1}$$

$$\begin{aligned}
T(n) &= a\, T(n/b) + O(n^d) \\
&= a^2\, T(n/b^2) + a \cdot O\left(\left(\frac{n}{b}\right)^d\right) + O(n^d) \\
&\;\;\vdots \\
&= a^k\, T\left(\frac{n}{b^k}\right) + \left[a^{k-1} O(b^d) + a^{k-2} \cdot O(b^{2d}) \cdots + a^0\, O(b^{kd})\right] \\
&= O\left(\sum_{j=0}^{k} a^{k-j} b^{dj}\right) = O\left(a^k \sum_{j=0}^{k} \left(\frac{b^d}{a}\right)^j\right)
\end{aligned}$$

$$n = b^k$$

Take $r = b^d/a$.

$r < 1$ ($\Leftrightarrow d < \log_b a$) : $O(a^k) = O(n^{\log_b a})$

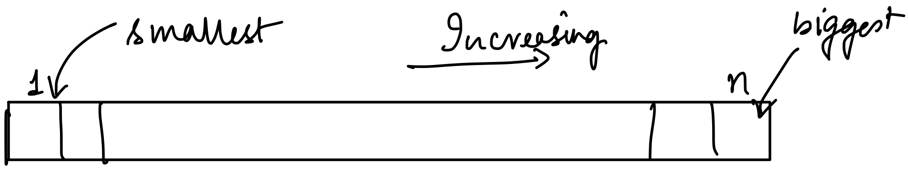
$r > 1$ ($\Leftrightarrow d > \log_b a$) : $O(a^k r^k) = O(n^d)$

$r = 1$ ($\Leftrightarrow d = \log_b a$) : $O(k \cdot a^k) = O(n^d \log n)$.

S is a list of n numbers. x is a given integer.
Your task: determine if $\exists$ i, j $(i \neq j)$ s.t. $S[i] + S[j] = x$. If they exist, determine them.

1. Sort S $\rightarrow$ $\Theta(n \log n)$.

2. After sorting, list is $A[1..n]$



look at $A_1 + A_n$. if $= x$ then ret $(1, n)$.

$T(n)$

if $< x$. then $A_1 + A_j$ (for some $1 < j < n$) is $\leq A_1 + A_n$. So such a thing can be ignored. So look at $A_j + A_n$ (for some $1 < j < n$) Move to $A_2 + A_n$.

$> x$. then ignore $A_j + A_n$ $\forall j$. Need to look at $A_1 + A_j$. Move to $A_1 + A_{n-1}$

This takes $O(n)$ time.

$n \log n$ "=" $n \log n + 1$ $\leq n \log n + T(n)$ "$\leq$" $n \log n + n$ "=" $n \log n$

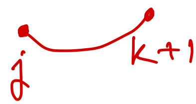
The entire algorithm takes $\Theta(n \log n)$

```
function   find_pair_sum (S, x) :

    S = sort(S)
    n = length(S)
    l = 1,  r = n,  flag = 0
    while (l < r):

            if (S_l + S_r = x) :
                  flag = 1
                  break

            else if (S_l + S_r > x) :
                  r = r-1
            else :
                  l = l+1
    if (flag = 0) :
            return (-1)
    else:
            return (l, r)
```

Hotels, penalty, $(200-x)^2$ . . .

$a_0=0 \quad a_1 \quad a_2 \quad a_3 \quad \ldots \quad a_n$

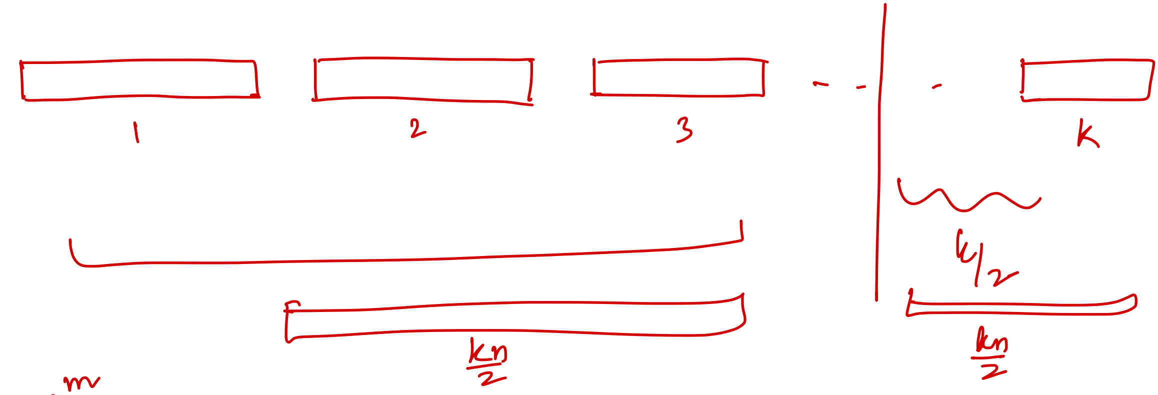
$S[0, 1, 2, \ldots, n]$



$S[0] = 0$

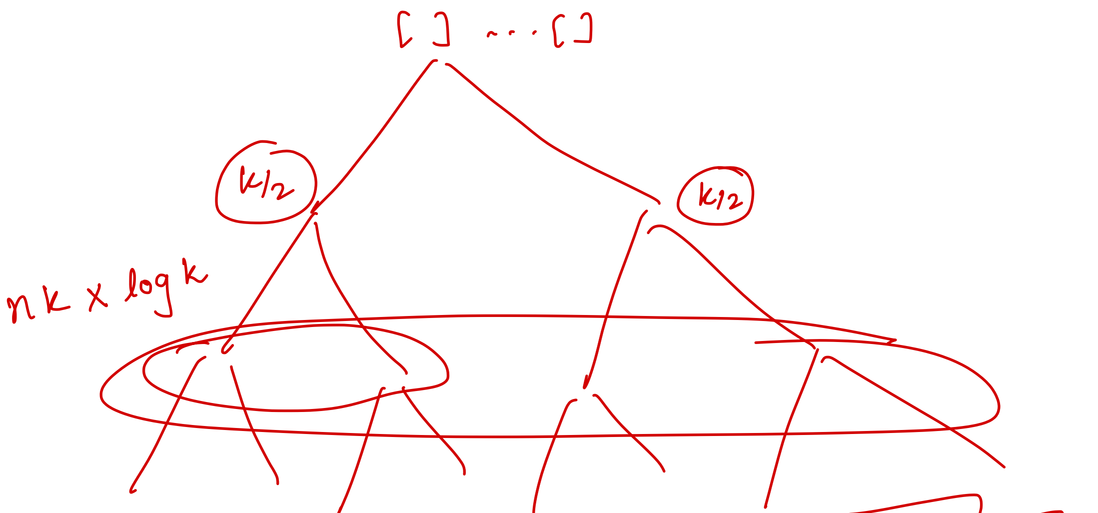
If you have $S[1], \ldots, S[k]$ then

$$S[k+1] = \min_{0 \le j \le k} \left[S[j] + \left(a_{k+1} - a_j - 200\right)^2 \right]$$



$k = 2^m$

$$\begin{cases} T(k) = T\left(\frac{k}{2}\right) \cdot 2 + O(kn) \end{cases}$$



nk/2 nk/2

$$T(n,k) = 2T\left(n, \frac{k}{2}\right) + O(nk)$$

$$= 4T\left(n, \frac{k}{4}\right) + O(nk) + O(nk)$$

$$\vdots$$

$$= 2^m T\left(n, \frac{k}{2^m}\right) + O(nk) \times m$$

$$= O(k) + O(nkm) = O(nk \log k)$$

$k = 2^m$