1 2 3 4
N=4 , (i=0; i< (1<<4); i++) ✓ 2^N

1011 → i 3

N = 4 → 1,2 w_1
→ 3,4 w_2

dp[i][w] → Consider the group of first i elements and assign them weight w

finally dp[n][W]

dp[i][w] ⇒ best value under this constraint.

dp[i][w] → Take ith element dp[i-1][w- wgh[i]] + val[i]
→ ignore ith element dp[i-1][w].

$V_i \le 1e^3$ $W \le 1e^9$, $N \le 100$

Max Ans → Max $V_i \times N$ = $1e^3 \times 1e^2 = \boxed{1e^5}$ $N \le 1e^2$

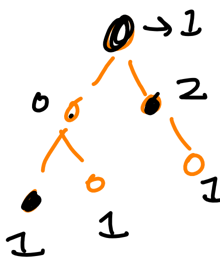
dp[i][v] ⇒ Can we achieve value v from first i elements? 0/1 ✓
✓ What is the minimum we weight from first i elements to get to v if it is possible.

V=1 → $1e^5$
dp[n][v] < w (knapsack size) ans = max(ans, v).

$dp[i][v] \rightarrow$ min weight.

Base Case: $dp[0][0] = 0$

Recursion ↳ $dp[i][v] = \min(\underbrace{wgh[i] + dp[i-1][v-val[i]]}_{\text{Take } i^{th} \text{ element}},\ dp[i-1][v])$



→ color this tree, every node gets colored white / black.

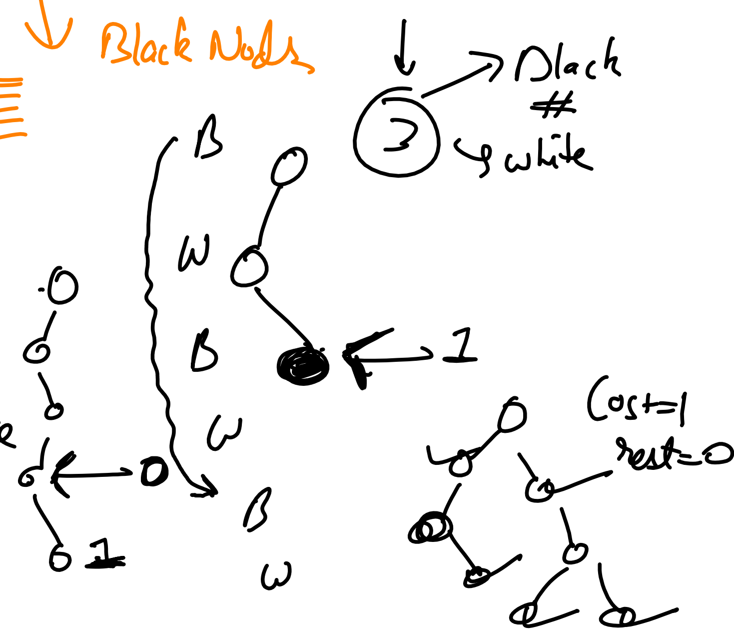
→ $cost[x] = |white[x] - Black[x]|$

white[x] ↓ Number of white Nodes in the path from root to x.

Black[x] ↓ Black Nodes

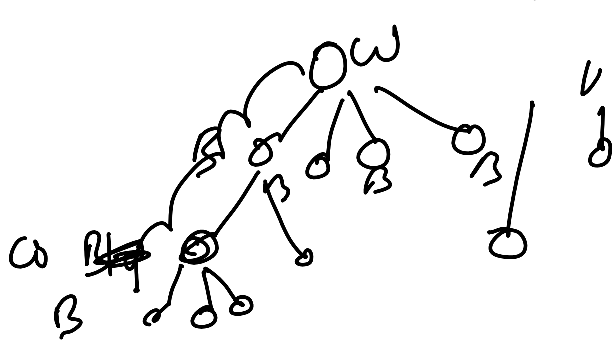
≡

Setup



1) Minimize the cost of the tree

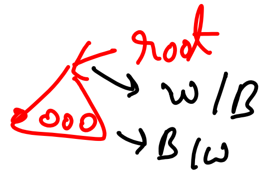
↳ Number of nodes with odd distance from root.

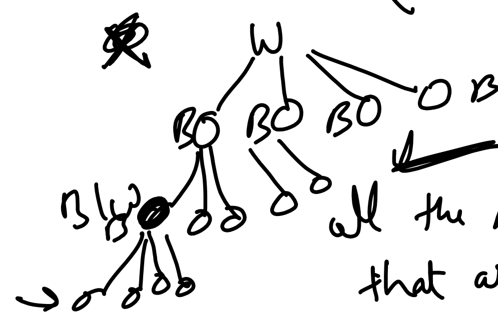


Value = abs (Count Black nodes - Count white Nodes).

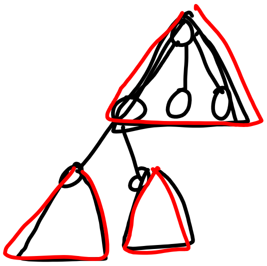
$d = 1$

↳ Maximize Value.





all the nodes that are at odd distance get to choose their color

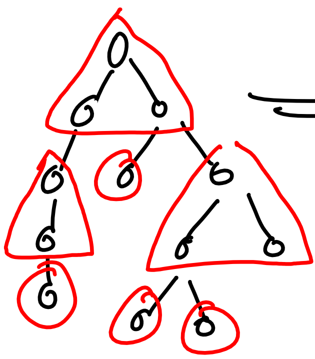


[(1, x) . . . . . .]

cnB += 1 and cnW += x

or

cnB += x and cnW += 1



(1, x) . . .

⟹ [(1,2) , (1,1) , (1,0) , (1,0) , (1,2) , (1,0) , (1,0)]

→ abs[cnB - cnW]

① dp[i][W][B] → works but out of constraint

↳ count of Black Nodes

↑ count of white Node

② W | B = Sum[i] - W → dp[i][W]. out of bounds.

(x,y) → (x-y)

↘ (y-x)

B - W

[↑

(x,y) (x,y)

Black - white

diff = 0 → diff += abs(x-y)

diff -= abs(x-y).

Minimize = abs(diff). + -1 abs(x-y)

[(x,y) . . . .] = [abs (x-y) . . . .]

Sum → +/-

finally we want abs (sum) to be minimum.

[i] [sum] → 0/1 $N^* O(N)$ N^2

1. Max unique value that array can have.

Sum ~ N

1 + 2 + 3 + 4 + 5 + 6 . . . ~ N

$\frac{n^*(n+1)}{2} \sim N$, $n \sim \sqrt{N}$

Max Uniq = $\sqrt{N}$.

for [i → 0 → $\sqrt{N}$]

~~for j=0 → count[i]~~

sum=0 → N

[ab(x-y)] → [(v, cnt v) . . .] size ~ $\sqrt{N}$

dp[i] [sum] $\sqrt{N}$

dp[i] [sum] ⇒ $\sqrt{N} * N$

(i, j)

5

Sum ~ O(N)

elements ~ O(N)

```
if(dp[i-1][j] >= 0)
    dp[i][j] = 0;
else if(dp[i][j - c[i]] >= 0 and dp[i][j - c[i]] < s[i])
    dp[i][j] = dp[i][j - c[i]] + 1;
else
    dp[i][j] = -1;
```

?

dp[i] [sum] → -1 Sum is not possible

→ min count of ith element required to get to this sum.

$\sqrt{N} * N$

3
4 7 8 →
↳ 4, (4, 3) , (4, 4)

2
2 7 →
2, (2, 2, 3) → 4

5 → abs(x−y) ≤ 1 in final array.
→ You can break elements.

1 Binary search
↓ ↳

min Binary
$x \longrightarrow y > x$

$[\,x \;\; x \; x \; x \;\; (x+1) \;\; (x+1) \;\; x \;\; x+1\,]$

$x = 1$

{1, 2, 1, 2 2 , 1, 2 3}
↓
1 / 2

a[i] = 7 → {2, 2, 2, 1}

8 → 2, 4, 2, 2

{x, x+1}

$a[i] \rightarrow \boxed{x, x+1}$

O(N)

(i) a[i] ≥ x

(x+1)₁ . . . . (x+1)_q

(ii) a[i] / (x+1) → q

a[i] % (x+1) → 0 → done (q)
→ x → done (q+1)

x+1 . . .

0 < rem < x

1
need = x − rem;

(x+1) . . . (x+1)

need ≤ q → (q+1) elements
otherwise → not possible

$\boxed{1 \le x \le 1e9}$ $x \to O(NX)$ = le11 ✗ ✓

IMP

$X \le \sqrt{1e9}$ ← Do it $O(N\sqrt{X})$ ✓

→ $X > \sqrt{1e9}$ → ????

$a[i] = 8$
$K = 3$

8/3 → 2
8%3 → 2

2 2 2
2 3 3

↙ (XN)

↘ what other thing to consider?

X

$a[i]$ ← K → $\sqrt{1e9}$ groups. → 1

-1 ← $a[i]/K$, $a[i]/K + 1$

$a[i] = 9$

[2 2 2 3 3 3 2 2 2] $K=3$

4 4 3 3 3 3 4 4

$K=3$
$a[i]=9$

9/3 → 3

3 3 3 → 2

x → $x+1$ → 4

↓ Groups

$\boxed{K < \sqrt{a_i}}$

$X \ge \sqrt{a_i}$

$x < \sqrt{a_i}$ ↙ Brute ↳ $\sqrt{x} * N$

$K > \sqrt{a_i}$

a_i

✗

[ ]

KN

K ✓

$k=1$

K

$\sqrt{N} * N$ $k < \sqrt{N}$ ✓

[ ]
[ ]
[
[

$\sqrt{N}\sqrt{N}$

$k > \sqrt{N}$

won't be more than $\sqrt{N}$ positions that's will update.

Q ✓

↳ (K, i) → $i, i+k, i+2k \ldots$ +2

↑

Single K → $O(N)$

$K = N$

✓

$(Q\sqrt{N} + N\sqrt{N})$

↑ $k > \sqrt{N}$ $k < \sqrt{N}$

$\sqrt{N}$ $\sqrt{N}$ $\sqrt{N}$

$\sqrt{N} \to \sqrt{N}$

segment(l, r)

$\sqrt{N}$ ✓