C1 Problem Review

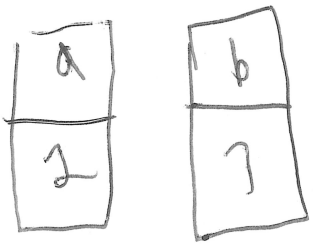
1. Cubbies - ANP
2. Galactic - Brygida
3. Next Square - ANP
4. Super Pond - Justin
5. Bandit - Justin
6. Runner - Justin
7. Smash Hit - Brygida
8. Errands - ANP
9. Polarize - Justin
10. Caves - Christian

Cubbies

x . . . x . . x . x . x . . . x
x x

Greedy Approach - left to right, use a cubby hole if possible. If you don't place a 'x' then your future placements are more constrained than mine.

Galactic



python : { } dictionary

Java Hashmap < String, Integer >

C++ unordered_map < string, int >

Next Square

Key Idea #1: floating pt ops inherent error

Key Idea #2: Binary Search

2615 $x > 0$

$x^2 \geq 2615$, $x \in$ Integer min (x)

low = 41, high = 60

Guess $x = 40$ $x^2 = 1600 \Rightarrow$ 40 is too small

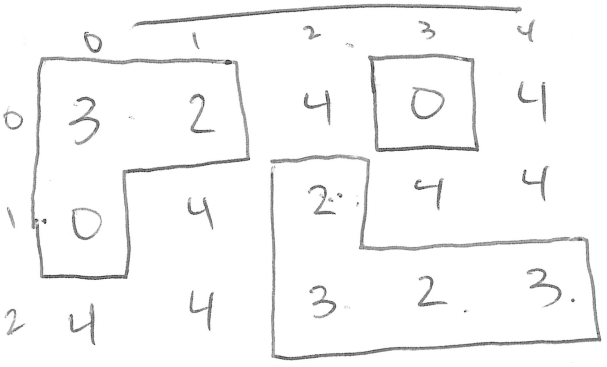
$x = 60$ $x^2 = 3600 \Rightarrow$ ~~60 is too big~~

real ans is 60 or less.

low = 0, high = 10^9

guess (low+high)/2 $\Rightarrow$ Use this to update low or high

Super Pond



res ← max volume crevice

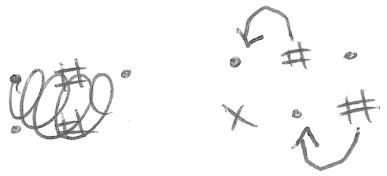
check each square {

if below surface level {

res = max(flood fill (1, 2), res)

v = 4 − h

Bandit



→ us losing

→ us winning

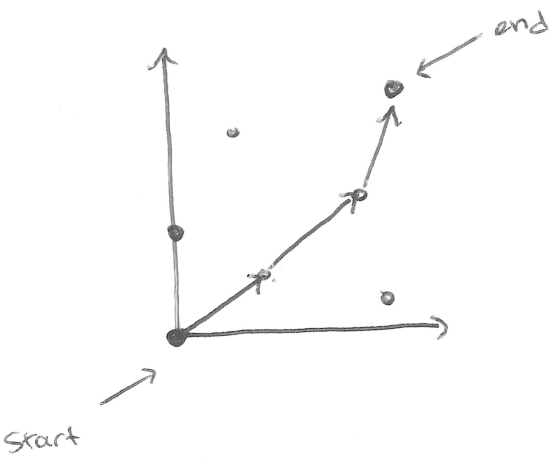
	0	1	2	3
0	0	(1)	2	3
1	1	2	3	4
2	2	3	4	(5)
3	3	4	5	6

each square has an x value, $X = i + j$

All squares with equal $(x \% 2)$ are even distances away from eachother.

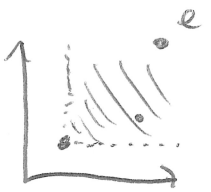
→ we win

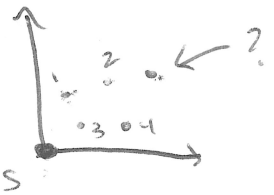
Runner



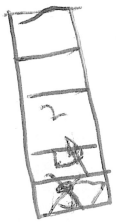
When traveling,

1) Move directly from pt a to pt b
2) ~~[illegible]~~ Not increasing distance from end point in both x & y dir.





Smash Hit



Errands

$n \leq 10$ errands

try all possible orders — permutation

some are not valid

1 -> 3

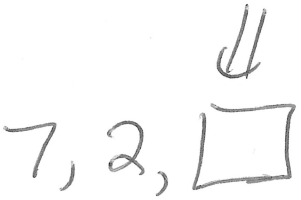
4 -> 6

7 -> 3

3 : 1, 7

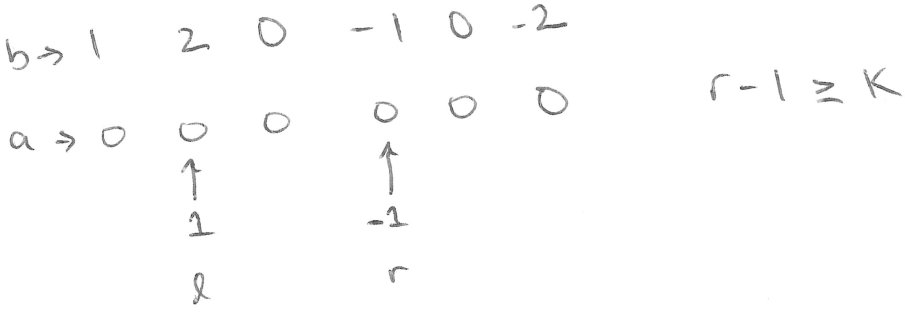
6 : 4

pre-req

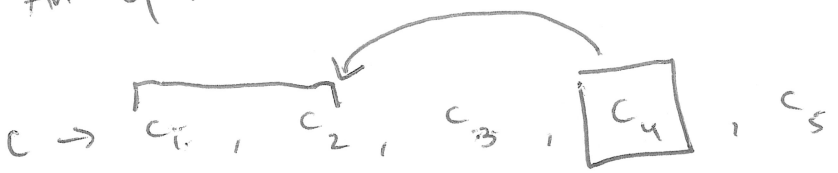


next item to place

```
for (int i = 0; i < n; i++) {
    if (used[i]) continue;
    if (!prereq(i)) continue;
    perm[k] = i;
    used[i] = true;
    res = min(res, recursivecall());
    used[i] = false;
}
```



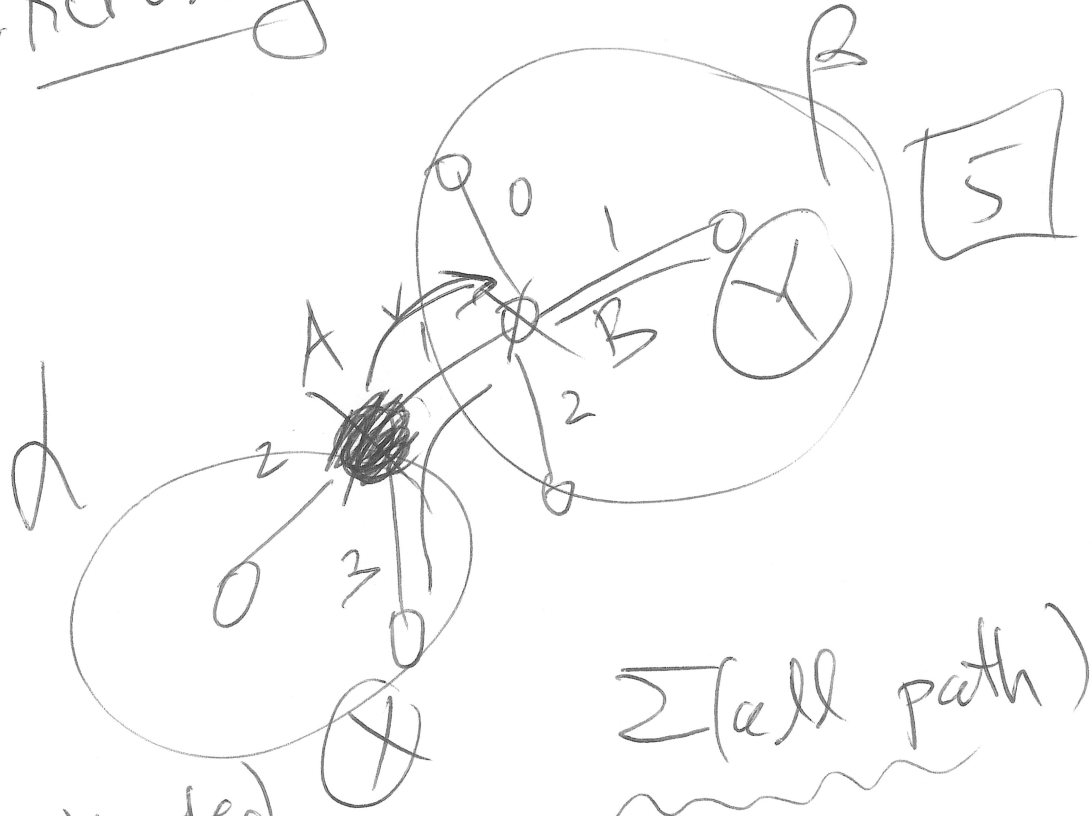
An operation has a net sum of 0




```
int canSpend = 0;
for (int i = 0; i < K; i++) {
    [illegible]
    if (b[i] is negative) return false;
}
for (int i = K; i < n; i++) {
    canSpend += max(0, b[i-K]);
    if (b[i] >= 0) continue;
    if (abs(b[i]) > canSpend) return false;
    canSpend += b[i];
}
```

* Centroid Decomposition

* Rerooting Justin (1:30 pm)



- Undirected
- Tree

Σ(all path)

BFS/DFS

n^2 ⟶ $n \log n$

$n \log^2 n$

n