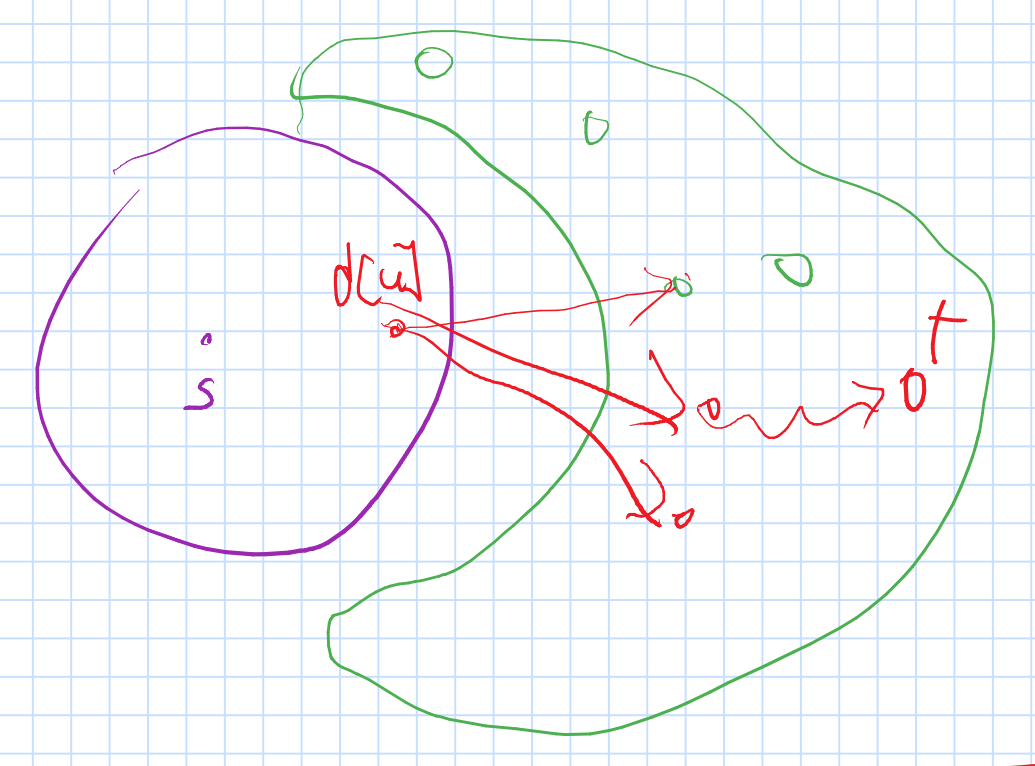
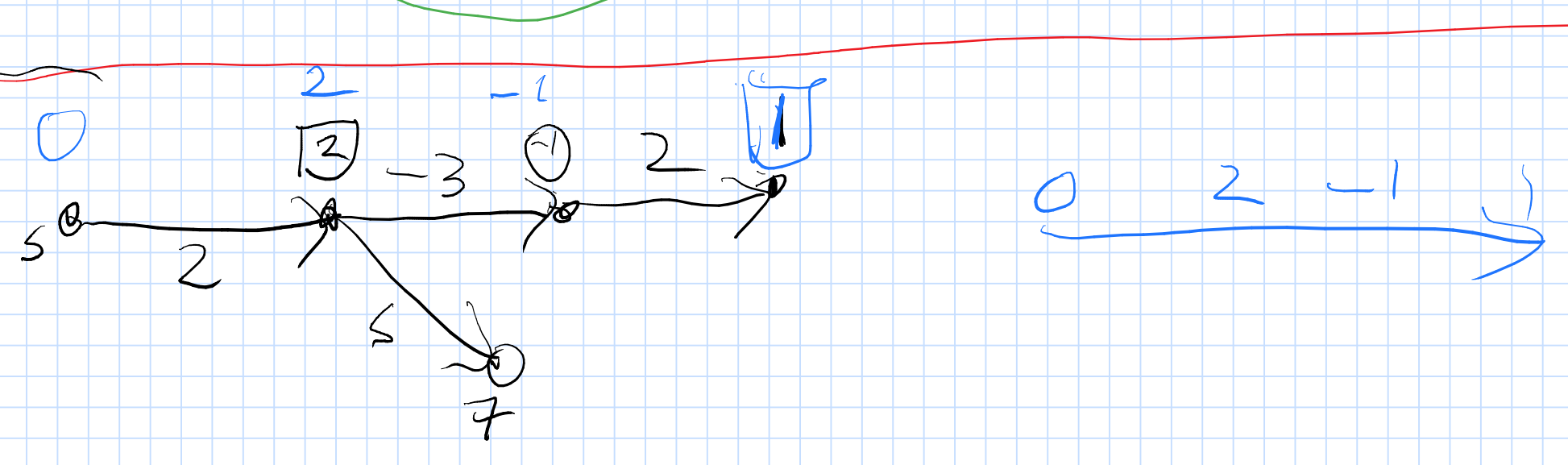
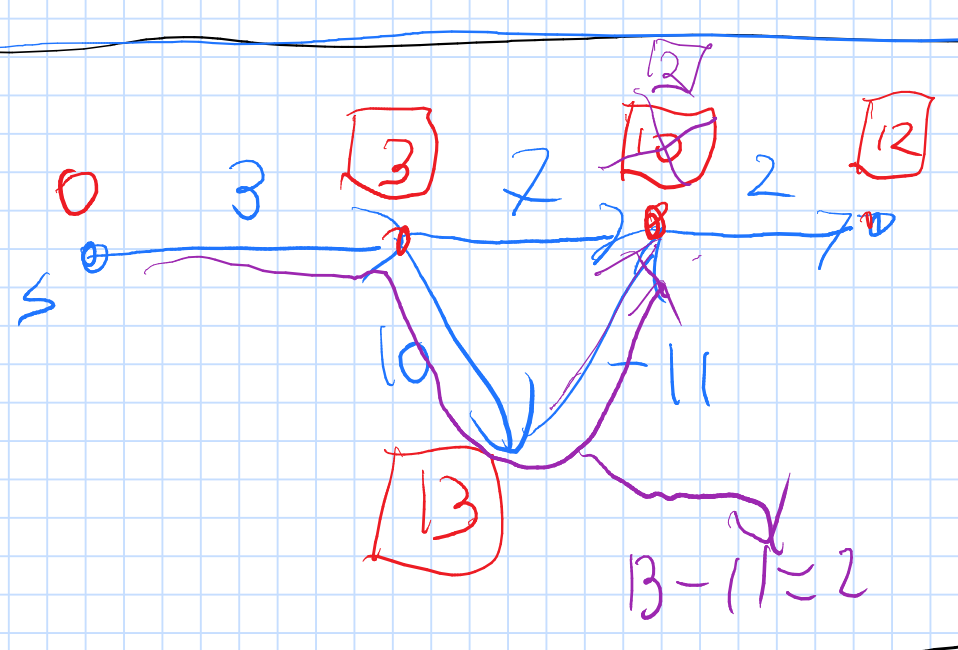
L19: shortest path II (negative weights)

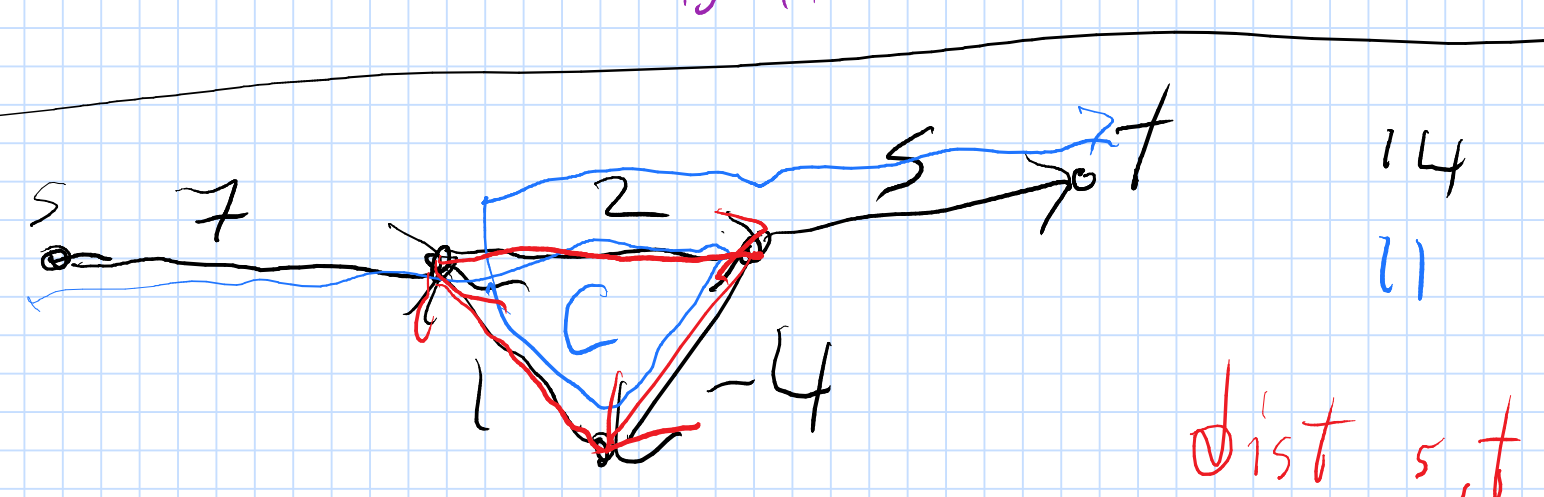


$O(n\log n + m)$

A*





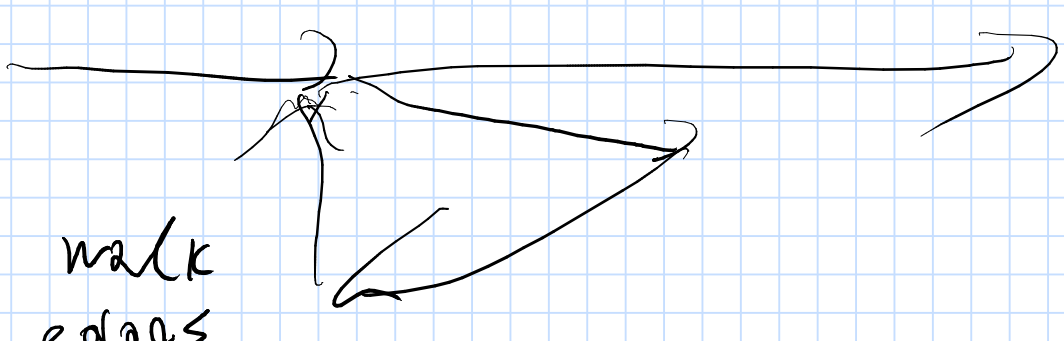


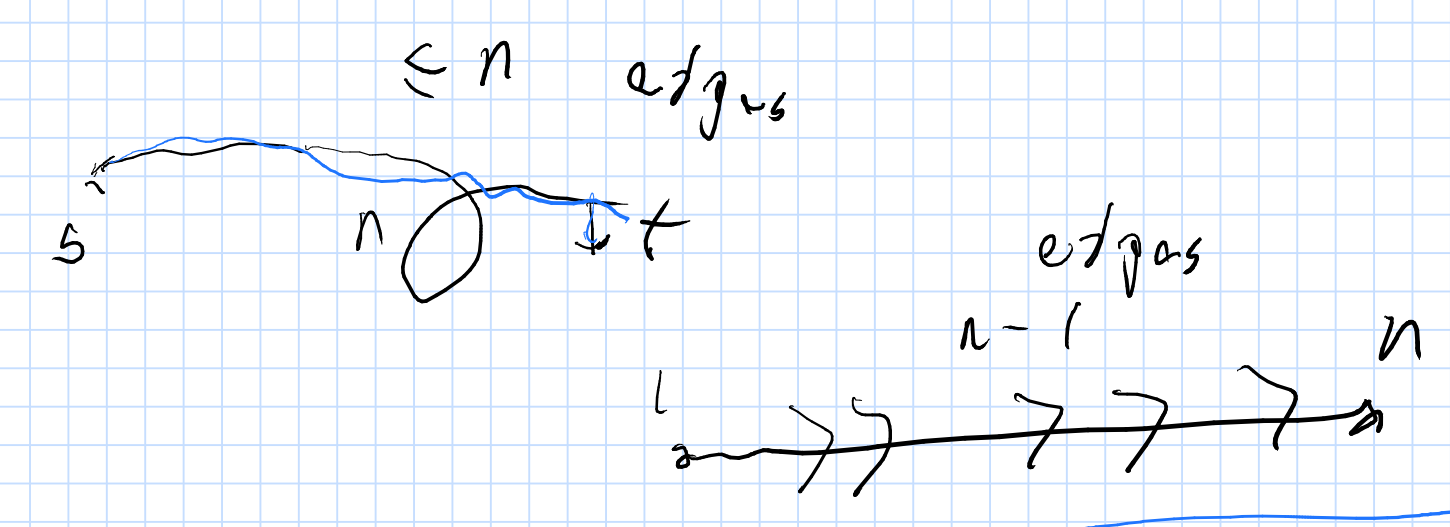
dist s,t going around negative cycle i times

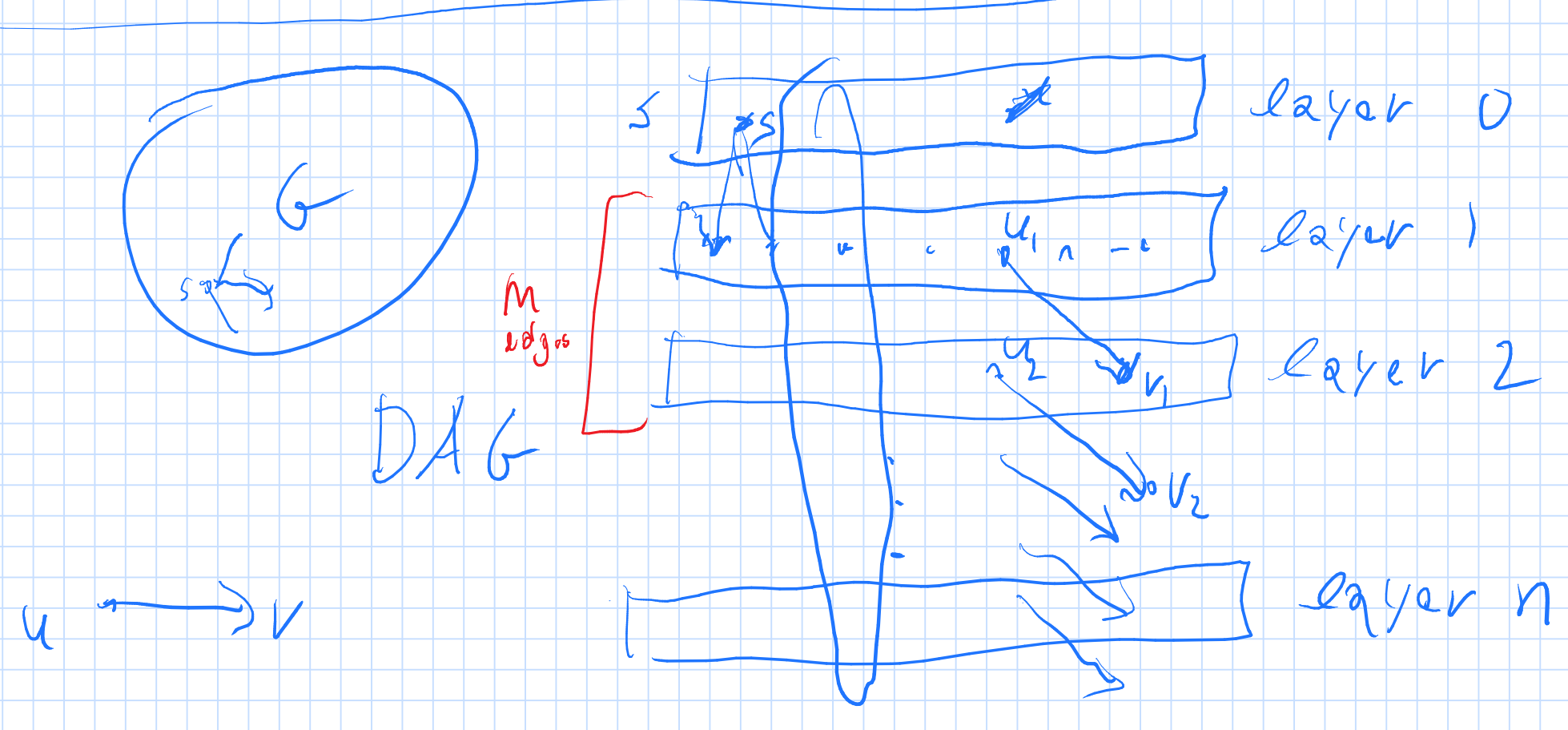
12 − i $\xrightarrow[i\to\infty]{} -\infty$

path ⇒ walk

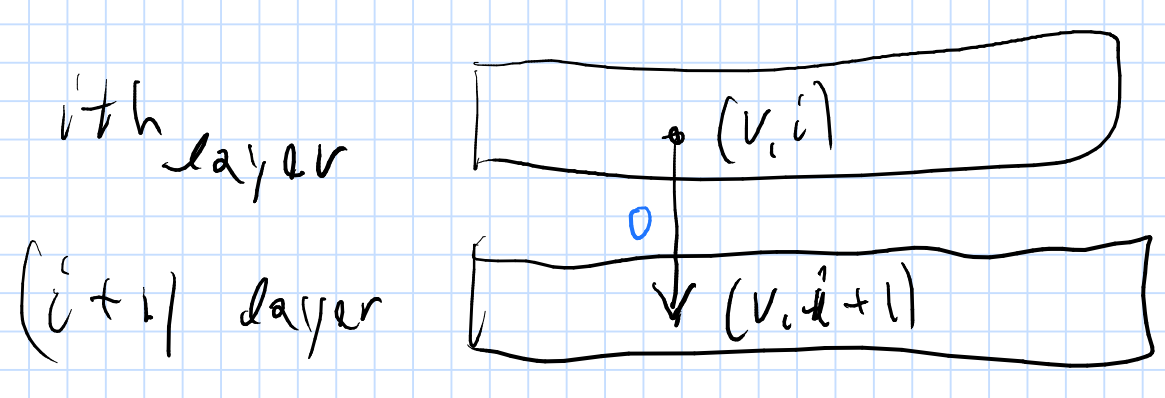
Compute for
k = 1, …, n
the shortest walk with k edges
from s to all the vertices in the graph.

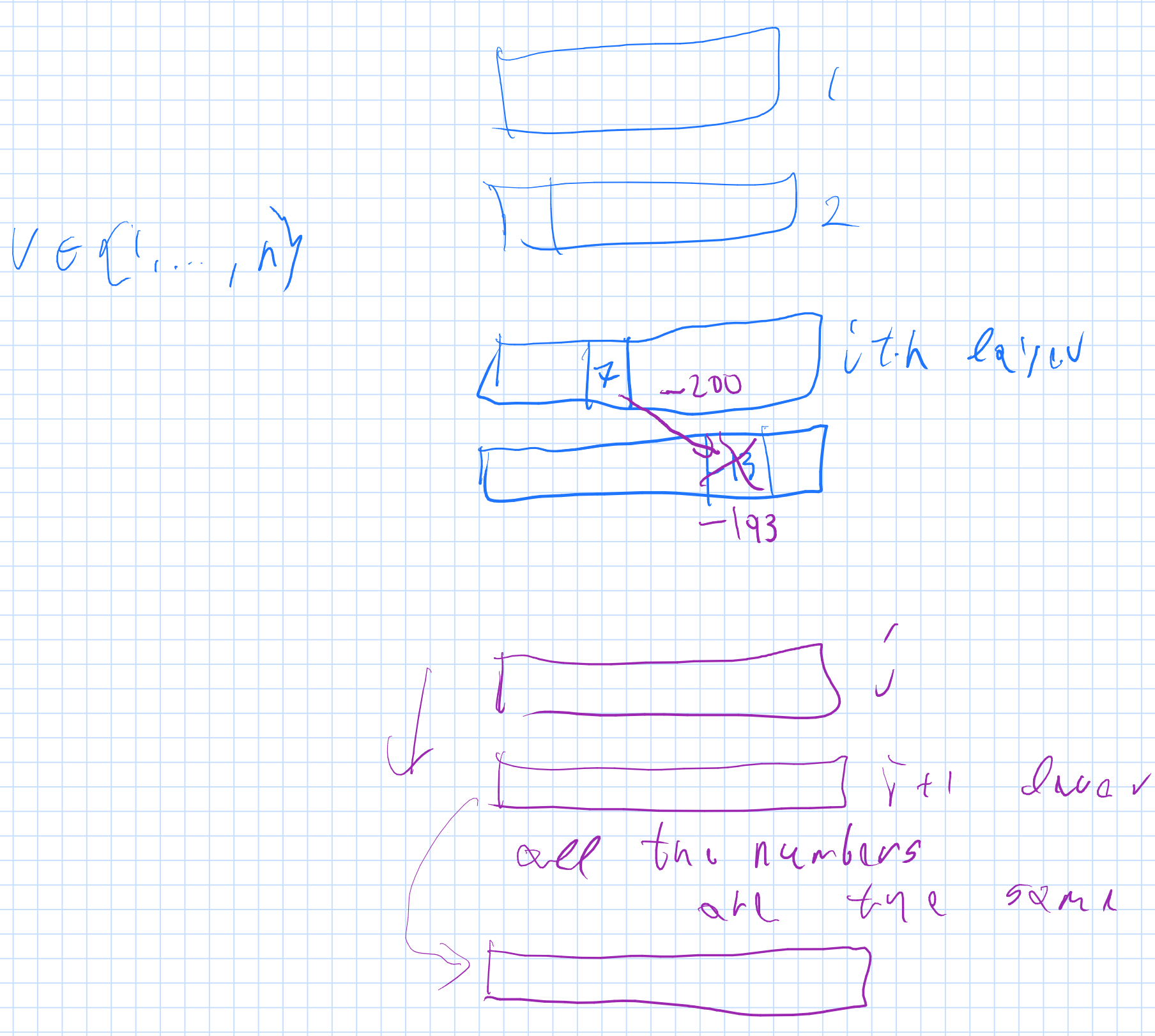


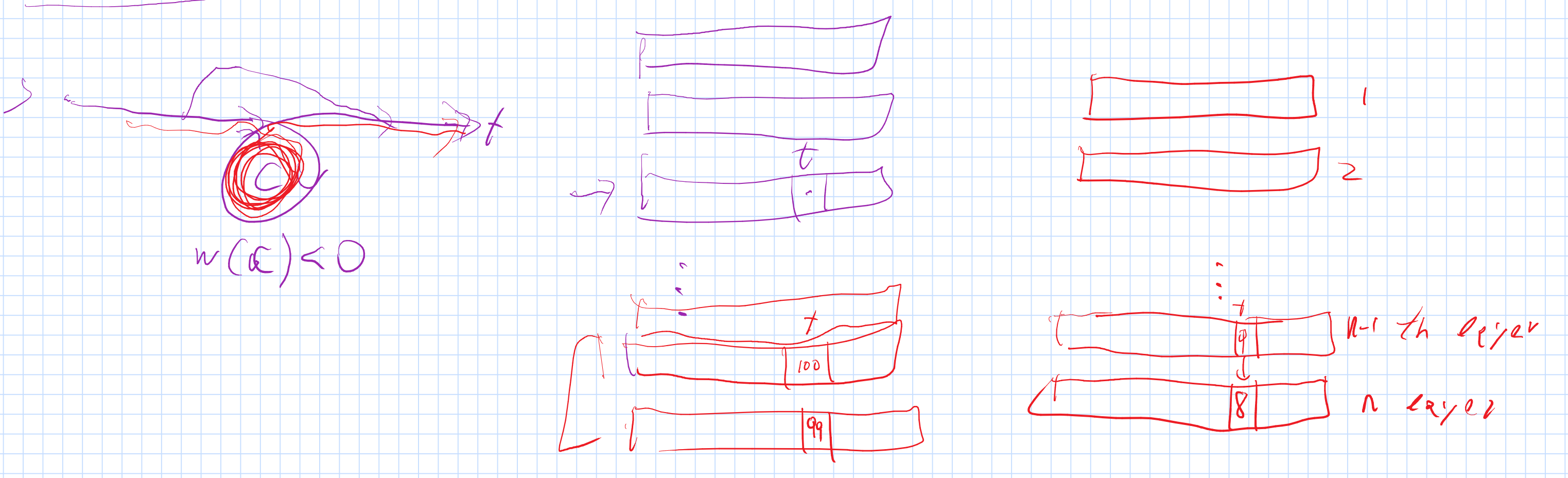


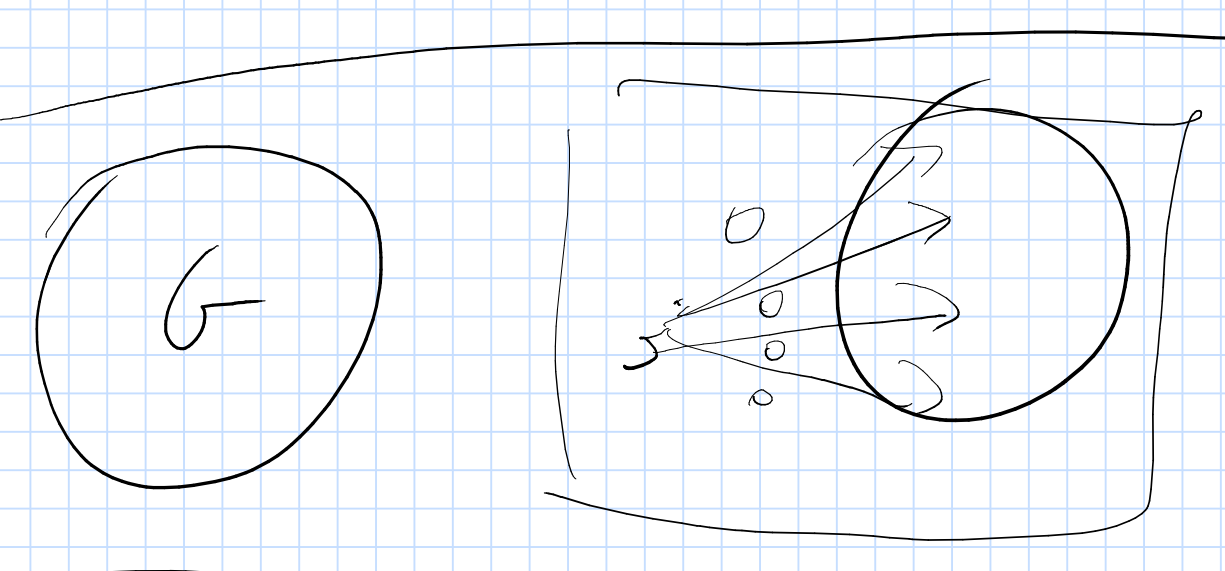


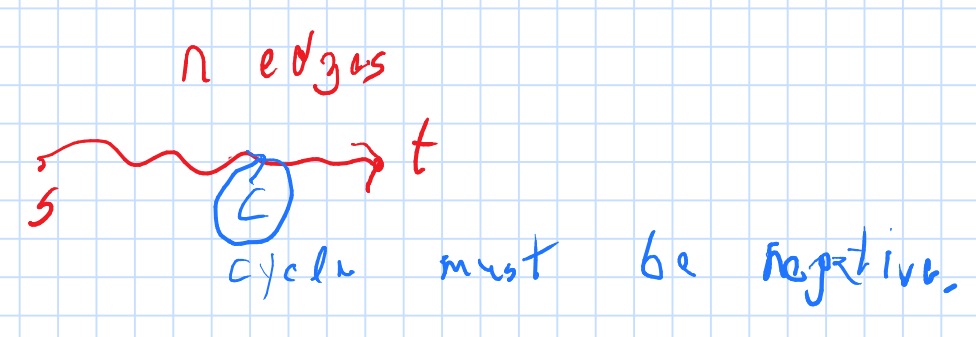
O(nm) Running time
Bellman-Ford algorithm



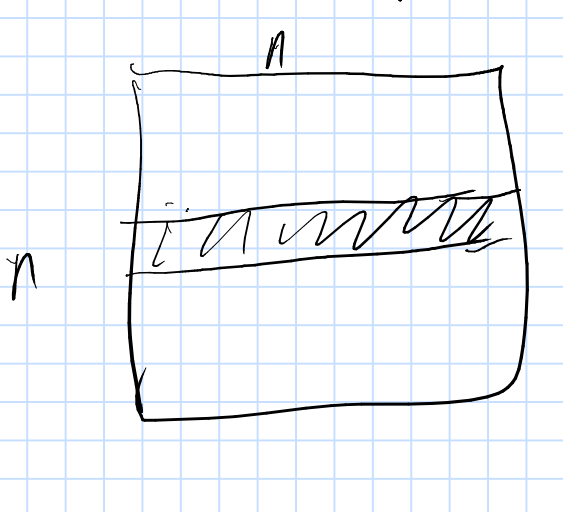






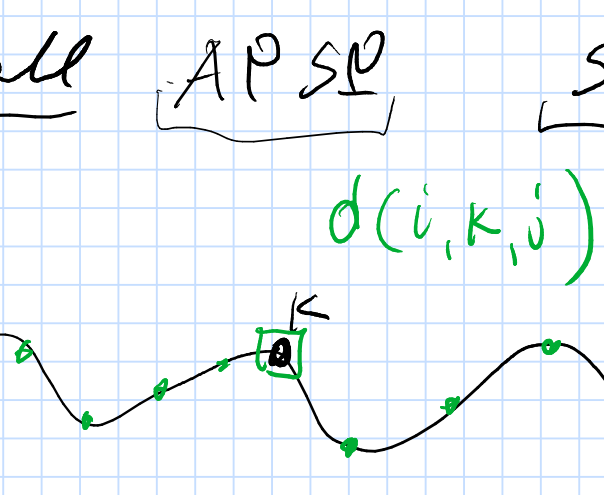


All pairs shortest path



$O(n\log n + m)$

$O(n^2\log n + nm)$



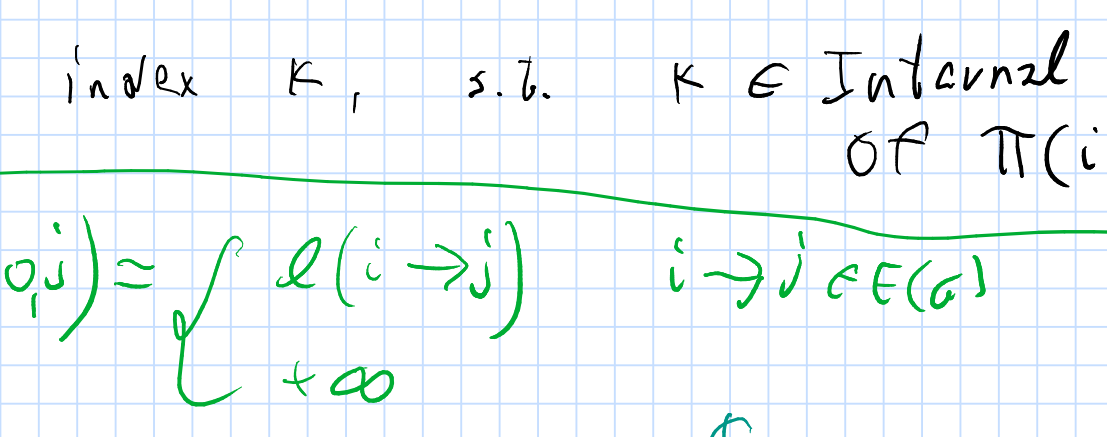
$O(nm)$

$O(n^2 m)$

$O(n^4)$

Floyd-Warshall APSP SSSP

$d(i,k,j) \equiv$ len of SP from i to j with max index of any middle vertex being k.



k maximum

1, …, n : number the vertices

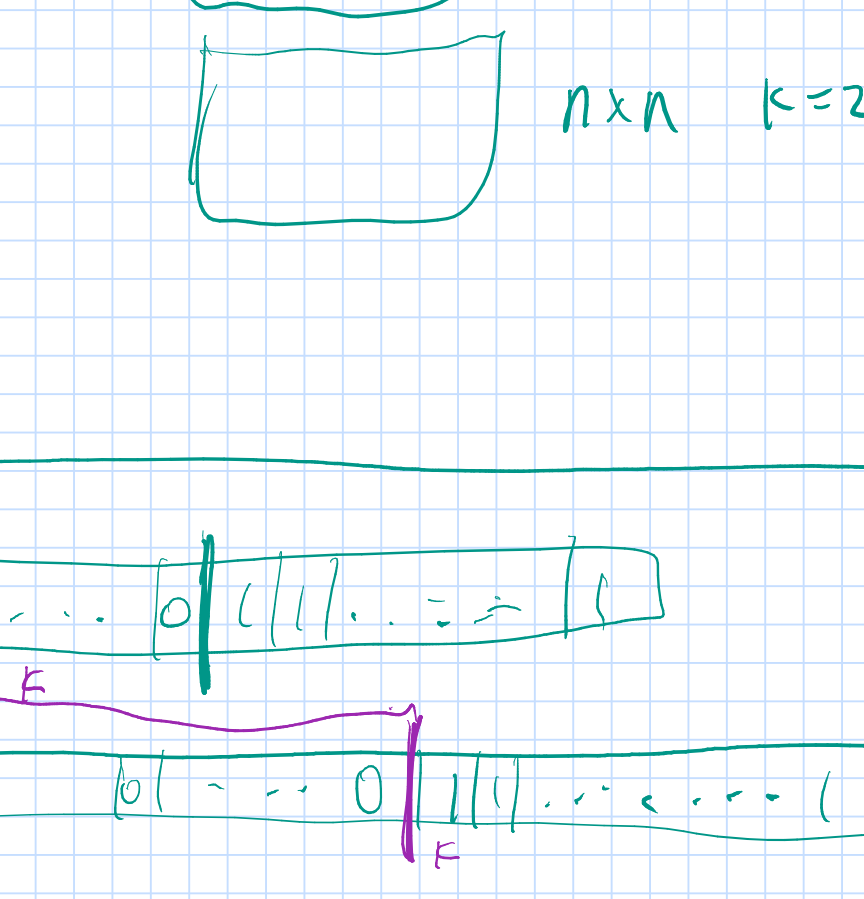
π(i, j) shortest path between i and j

max index k, s.t. k ∈ Internal vertices of π(i, j)

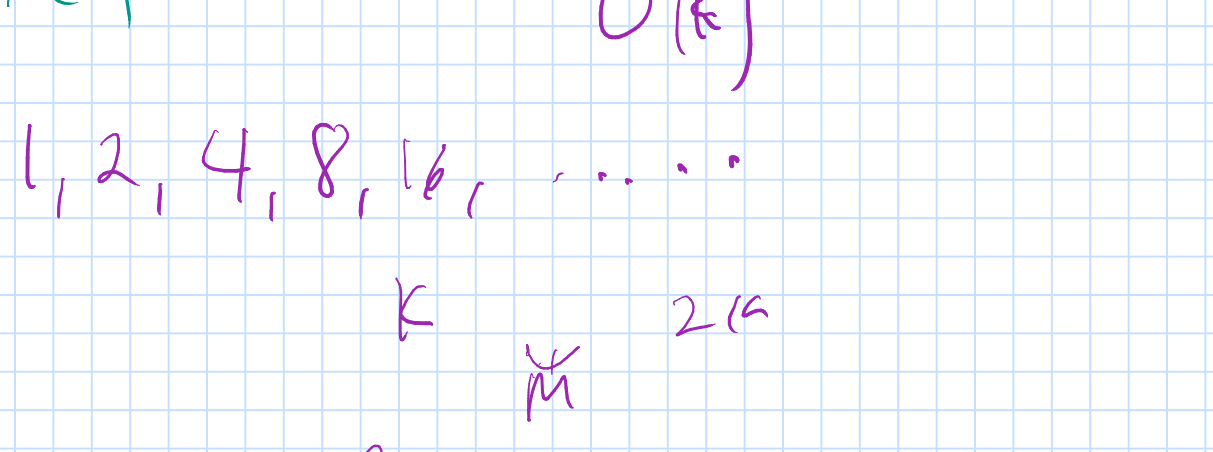
$d(i,0,j) = \begin{cases} \ell(i\to j) & i\to j \in E(G) \\ +\infty \end{cases}$

$d(i,k,j) = \min \begin{cases} d(i,k-1,j) \\ d(i,k-1,k) + d(k,k-1,j) \end{cases}$ O(1)

$O(n^3)$



$O(n^3)$



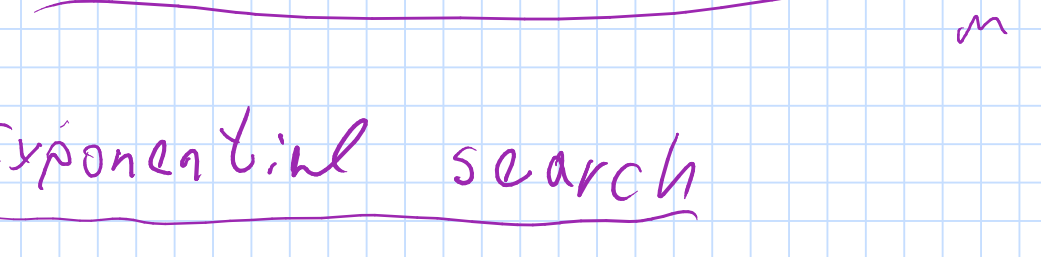
f(i)

O(k)

1, 2, 4, 8, 16, ……

k 2k

f(m) = 1



binary search

Exponential search