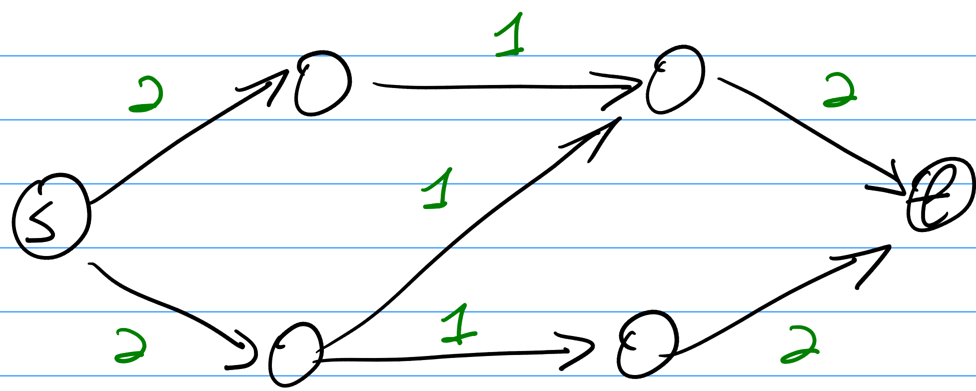
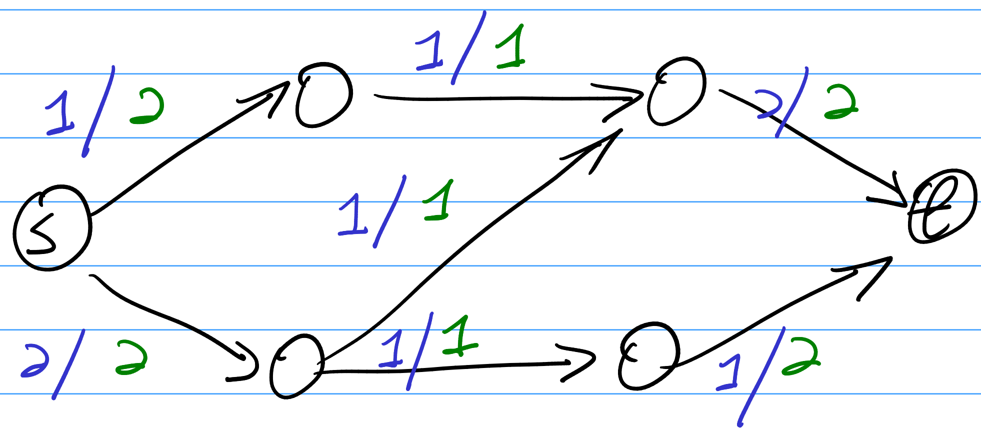
Topic 7: Network Flow

Min-Cut, Min-Cost

Network Flow (Max-Flow)





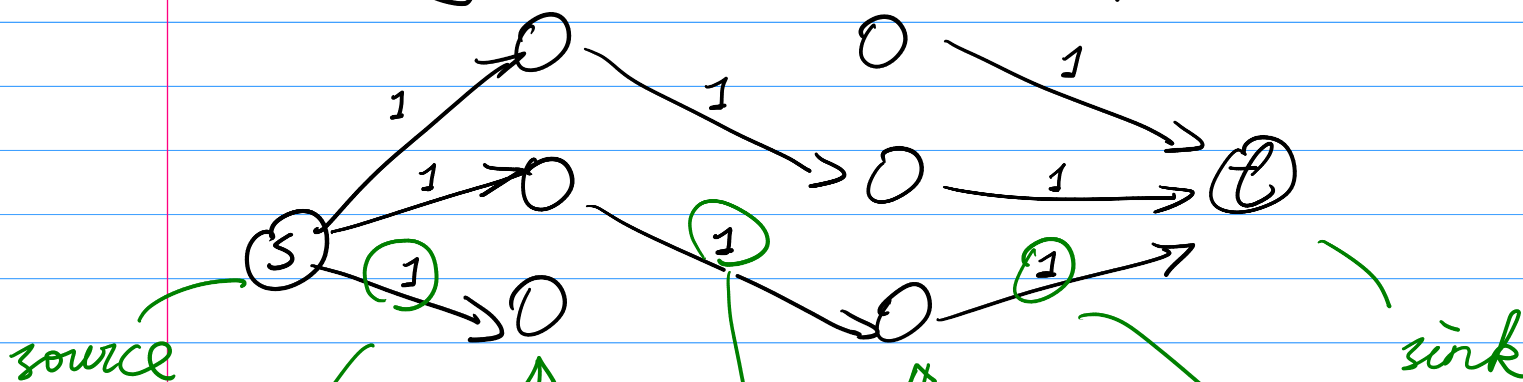
Matching Problem (Bipartite)

There are n men and m women.

Based on their preferences only certain pairs of them are compatible ($M[i][j] = 1$ iff i-th man and j-th woman are compatible)

Find the maximum numbers of pairings possible.

Matching Problem (Bipartite)



Each man contributes 1 to the pairing

Each man

Each woman

Each woman contributes 1 to the pairing

It is possible to match the two for 1 pairing.

Algorithms for Network Flow.

Consult your reference.
(Dinic. ISAP)

Complexity : Magic

Really bad on arbitrary graphs ($O(V^2E)$ for Dinic)
Magically becomes better on certain graphs
(c.f. Hopcroft-Karp) ← (Dinic is $O(\sqrt{V}E)$ for matching prob graph)

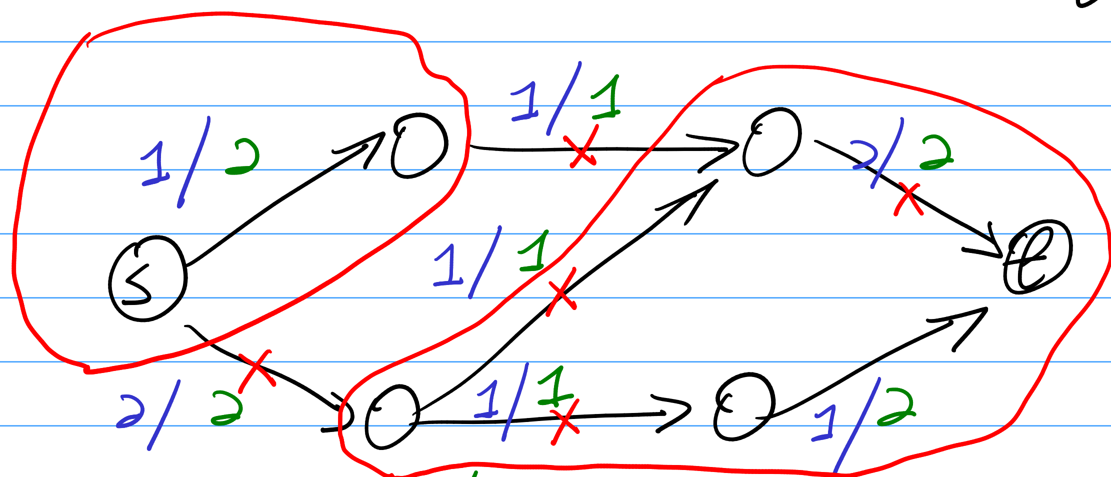
Result : Try different algorithms and pray that one does not TLE.

Magic Potion

Min-Cut

Definition: Minimum set of edges to delete s.t. S is not connected to T

Max-Flow Min-Cut Theorem: max-flow = min-cut



Remove every edge w/ full flow.

Then nodes reachable from S form a cut.

Min - Cut

1	10	2
3	4	2
8	2	8

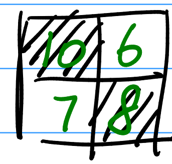
Square Number-Picking:

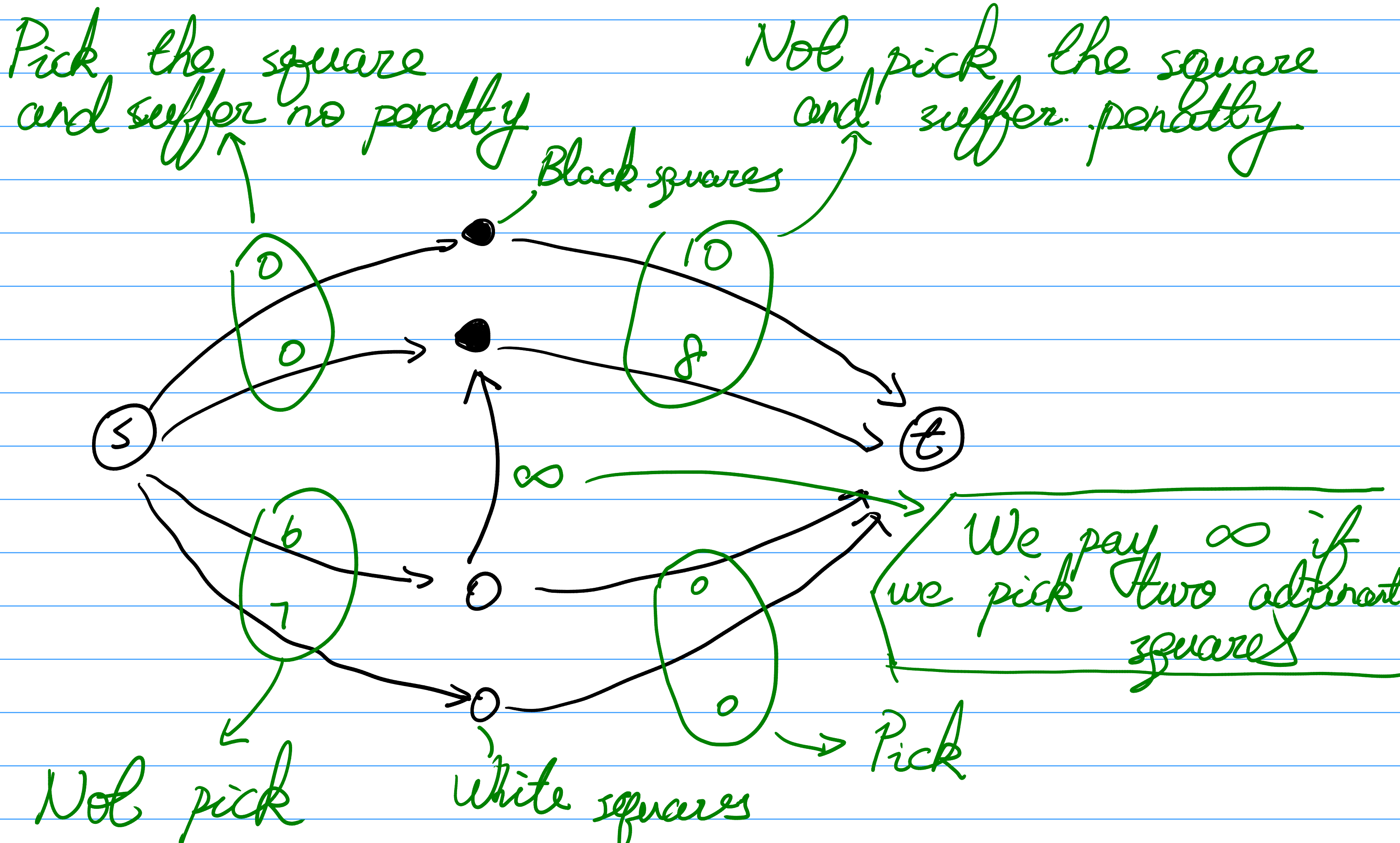
Find the set w/ maxium sum s.t. no two squares touch each other.

We encode 'cutting an edge' as 'making a decision' for the problem.

Min - Cut

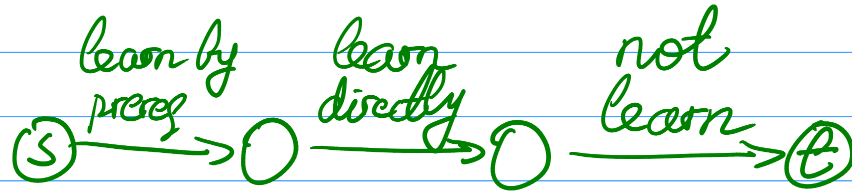
Two-color the squares



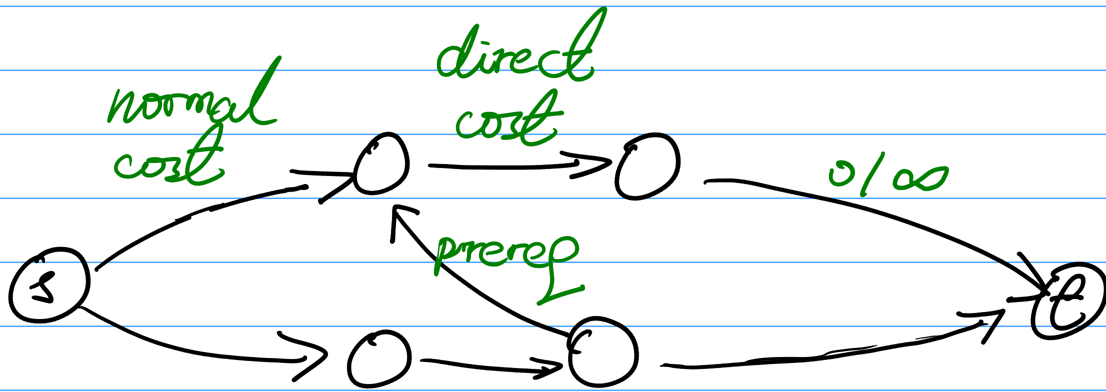


Kejin Game

Kejin Game

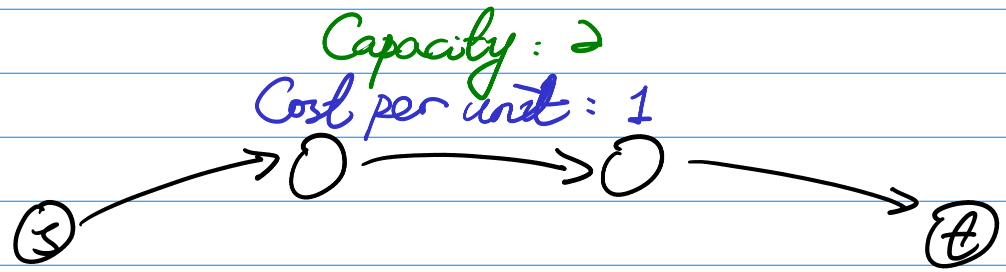


learn by prereq
learn directly
not learn
s
t



normal cost
direct cost
0/∞
prereq
s
t

Min-Cost Flow



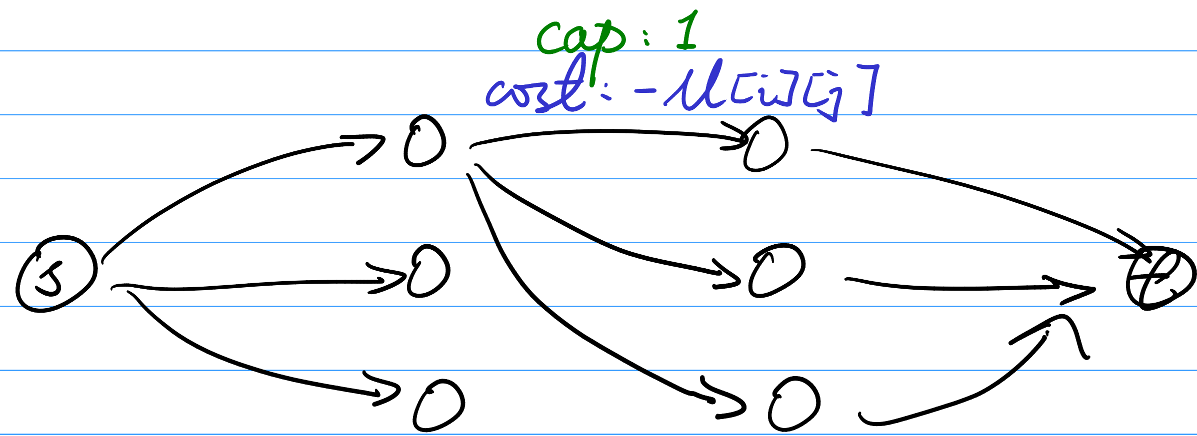
Given max flow. find min cost for the max flow

Algorithms: EK, ZKW, etc.
(see reference)

Weighted Matching

If we match the i-th man and j-th woman,

we gain M[i][j] money.



Coding Contest

Hiring Employees

Hiring Employees (Temporal Graph?)