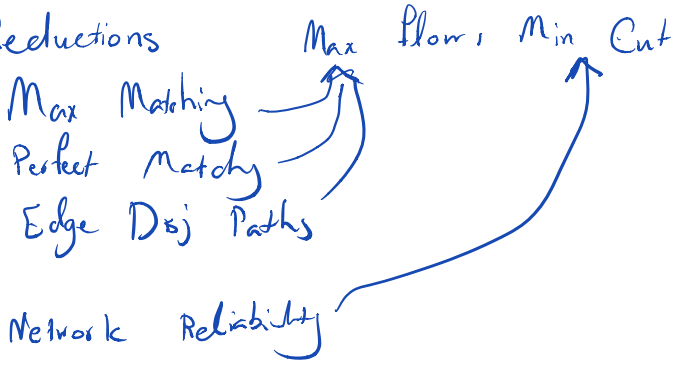
Reductions



Reduce to Mincut when we need to find a 2-parti of vertices.

Another Theme: Sometimes you need to construct graph.

when to use → DP: useful when a natural ordering on subsets of prob.

→ Max-flow, Mincut: No ordering is involved.

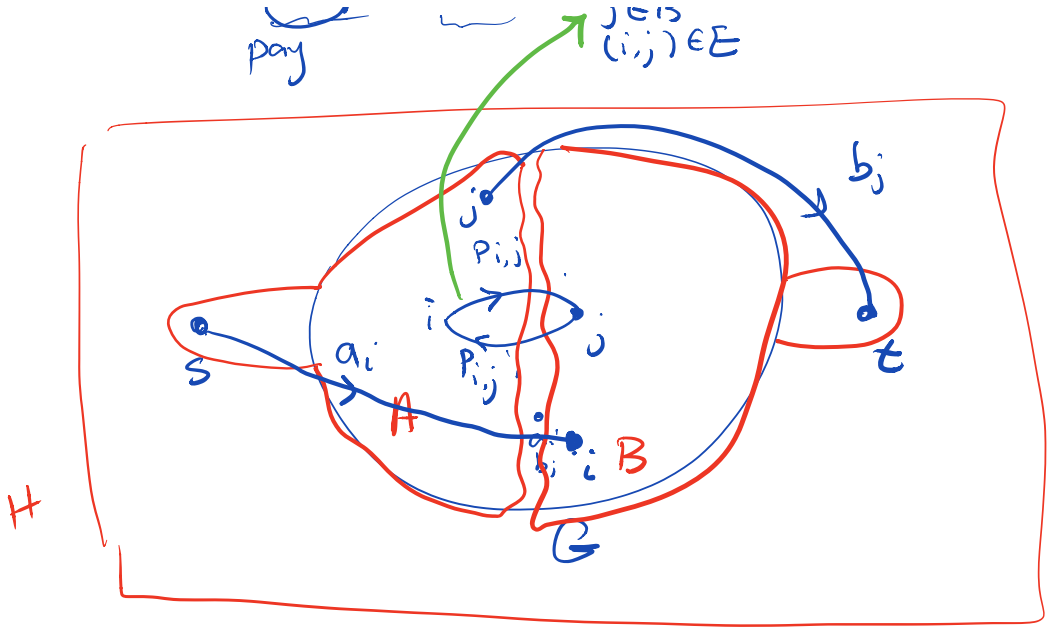
$$\max_{(A,B)} \sum_{i \in A} a_i + \sum_{j \in B} b_j - \sum_{\substack{i \in A \\ j \in B \\ (i,j) \in E}} p_{i,j}$$

does not change OPT (A,B)

$$= \min_{(A,B)} -\left(\sum_{i \in A} a_i + \sum_{j \in B} b_j - \sum_{\substack{i \in A \\ j \in B \\ (i,j) \in E}} p_{i,j}\right) + \underbrace{\sum_{i \in V} a_i + \sum_{j \in V} b_j}_{\text{only depend on input not } A,B}$$

doesn't depend on A,B so doesn't change OPT (A,B) partition

$$= \min_{(A,B)} \sum_{i \in B} a_i + \sum_{j \in A} b_j + \sum_{\substack{i \in A \\ }} p_{i,j}$$

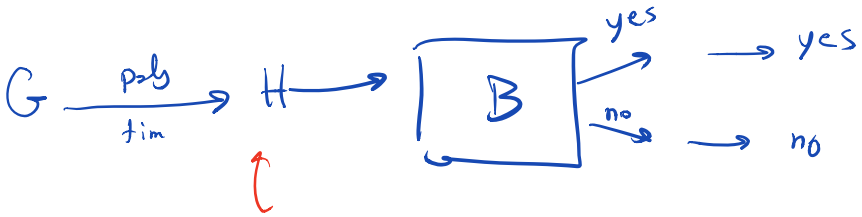


connect s to all i with edge of capa a_i

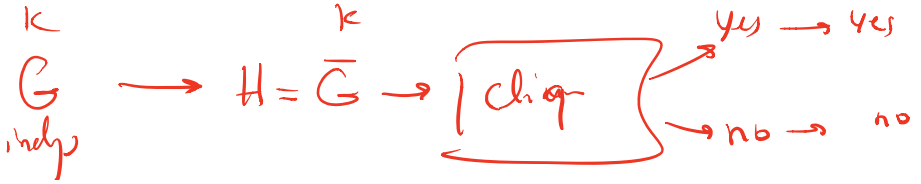
connect all j to t " " " " b_j

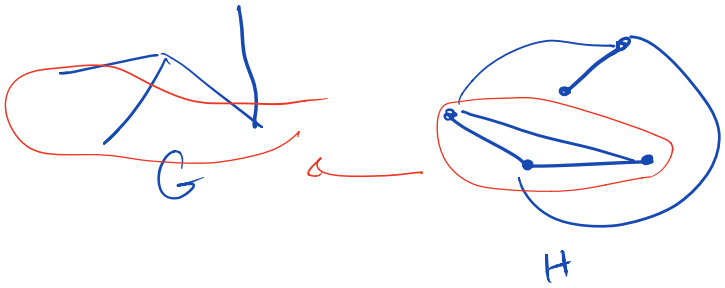
$A \leq_p^{1'} B$

G



main diff par is to construct H.




G
H