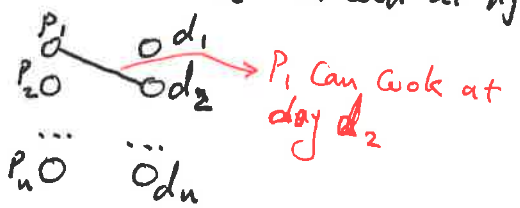
1-a) (10 pts)

Left nodes: $P_1, P_2, \dots, P_n$

Right nodes: $d_1, d_2, \dots, d_n$

$(P_i, d_j) \in E$, if P_i can cook at d_j



* There is a feasible scheduling iff

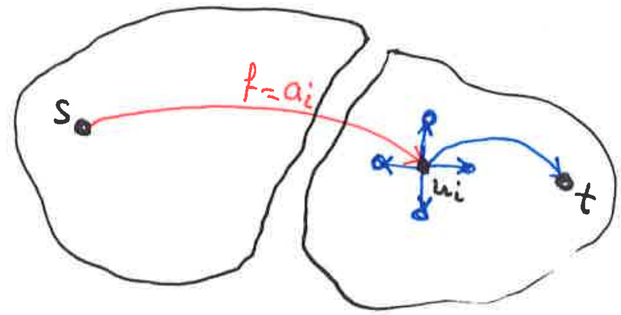
1-b)

- remove the two people that are assigned to the same day
- Construct the network flow by adding the Source (S) and dest. (t) nodes; Connect all nodes P_i to S with Capacity 1; Connect all nodes d_i to t with Capacity 1; add Capacity ∞ to all edges b/w P_i and d_j
- for all edges (S, P_i) where P_i is matched, Set the flow as 1; for all edges (d_j, t) where d_j is matched, Set the flow as 1; for all matchings (P_i, d_j), Set the flow as 1.
- Run the F-F Algorithm to Find the matching!

* Since the Current flow is $(n-2)$ and the max flow is $\underline{n}$, the algorithm runs for $\underline{2}$ iterations. every iteration is in $O(m) \leq O(n^2)$

$\Rightarrow$ The Alg runs in $O(n^2)$

2)



- The max Capacity of each blue edge is $\underline{\underline{d}}$
- Currently the whole Capacity of (S, u_i) has been exhusted
- we want to make Sure, no matter what this edge will not be exhusted
  - if So, u_i will belong to the Cut of S.
- The Max out flow from u_i is $\underline{\underline{5d}}$

$\Rightarrow$ if we increase the Capacity of (S, u) by $5d+1$, this edge will never be in the t-Side

$\Rightarrow$ This is what we do

after the increase the algorithm will repeat for at most $5d+1$ iterations

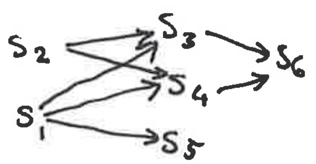
$\Rightarrow O(d(m+n))$, $m = O(n)$

$\Rightarrow O(dn)$ update time

3)

observation:

Constructing the *subset* graph between the sets $S_1 \cdots S_n$, like the following example:



we need at most $\underline{n}$ permutations but, following each edge, we can reduce one permutation.

for example, for $S_2 \to S_3$ we can construct the permutation

$\boxed{S_2 \cap S_3 \;|\; S_3 \setminus S_2}$

* The problem is that we can't choose two edges with the same end node

because for example for $S_2 \to S_3$, $S_1 \to S_3$ we either need the permutation according to S_1 or S_2.

$\Rightarrow$ we construct a bipartite graph that has the sets in its two sides. There is an edge from S_i to S_j if $S_i \subset S_j$

* Adding the S and t nodes we construct the network flow and set all edge weights to $\underline{1}$.

* Let x be the max-flow. we need $(n-x)$ permutations.

$\Rightarrow$ if $(n-x) \le l$, the problem has a valid solution.

4)

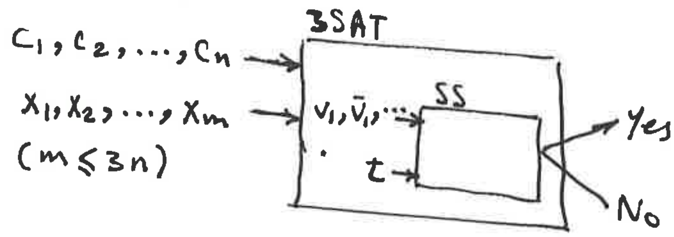
(A) Show $SS \in NP$.

Certificate: given a subset of numbers

Verification: does those add up to $\underline{t}$? it can be done in $O(n)$ ✓

(B) Reduction [CLRS]

$3SAT \le_p SS$



	x_1	x_2	...	x_m	C_1	C_2	...	C_n
v_1	1	0	...	0	0	0	...	1
$\bar{v}_1$	1	0	...	0	1	0	...	0
v_2	0	1	...	0	1	0	...	0
$\bar{v}_2$	0	1	...	0	0	0	...	0
⋮	⋮	⋮	...	⋮	⋮	⋮	...	⋮
v_m	0	0	...	1	0	0	...	0
$\bar{v}_m$	0	0	...	1	0	1	...	0
C_{11}	0	0	...	0	1	0	...	0
C_{12}	0	0	...	0	2	0	...	0
C_{21}	0	0	...	6	0	1	...	0
C_{22}	0	0	...	0	0	2	...	0
⋮	⋮	⋮	...	⋮	⋮	⋮	...	⋮
C_{n1}	0	0	...	0	0	0	...	1
C_{n2}	0	0	...	0	0	0	...	2
t	1	1	...	1	4	4	...	4