Introduction: Complete proof of induction (slide 30)

Induction hypothesis: $\sum_{i=0}^{n} a^i = \frac{a^{n+1}-1}{a-1}$.

Induction step:

$$\sum_{i=0}^{n+1} a^i = \sum_{i=0}^{n} a^i + a^{n+1} = \frac{a^{n+1}-1}{a-1} + a^{n+1} = \frac{a^{n+1}-1}{a-1} + \frac{a^{n+2}-a^{n+1}}{a-1}$$

$$= \frac{a^{n+2}-1}{a-1}$$

Stable Matching:

A slowest execution of TMA.

In class we argued that the Traditional Marriage Algorithm will never require more than n^2 days to terminate. In fact, this is not tight and the actual bound is $n^2 -2n+ 2$ (maximum number of days until the algorithm terminates). I challenged you to describe a set of preference lists that requires $n^2 -2n + 2$ days to terminate.

Solution:

Here are possible lists (there are other solutions):

Boys:
b1: 1, 2, 3, …., n-1, n
b2: 2, 3, 4, …., n-1, 1, n
In general, for k=1 to n-1 the list of bk is: k,k+1,…,n-1,1,2,…,k,1,n
For the last boy:
bn: 1, 2, 3, …, n-1, n

Girls:
g1: 2, 3, 4,….,n,1
g2: 3, 4, 5, ….1,2
g3: 4, 5, 6,…..2,3

In general, for k=1 to n-1 the list of gk is: k+1,…,n,1,2,…,k
For the last girl any list can do. She never rejects.

Every day exactly one boy is rejected, (n-1) boys are rejected (n-2) times, one boy (the one that ends up with girl n) is rejected n-1 times. In one additional day no one is rejected.
All together (n-1)(n-2)+(n-1)+1 = n^2-2n +2 days are required.

Scheduling:

Theorem 1: The following objective functions are equivalent (a schedule that is optimal for one is optimal also for the others):

(i) Min C_{max} (makespan)

(ii) Max Avg(N_p) (average # of processed jobs)

(iii) Min $\sum_k I_k$ (total idle time)

Proof :

(i) ≡ (ii) Note that Avg(N_p) = $\sum_j p_j / C_{max}$ This holds since job j is processed p_j time units along [0, Cmax] therefore it contributes p_j / C_{max} to Avg(N_p). The value of the numerator is independent of the schedule, therefore maximizing Avg(Ng) is equivalent to minimizing the denominator C_{max} .

(i) ≡ (iii) The idle time of machine k is Ik = Cmax - $\sum_{\{j|\text{assigned to Mk}\}} p_j$. Therefore, $\sum_k I_k = mC_{max} - \sum_j p_j$. m and $\sum_j p_j$ are independent of the schedule, therefore minimizing $\sum_k I_k$ is equivalent to minimizing the C_{max} .

Theorem 2: The following objective functions are equivalent (a schedule that is optimal for one is optimal also for the others):

(i) Min $\sum_j C_j$ (completion time)

(ii) Min $\sum_j F_j$ (service time)

(iii) Min $\sum_j W_j$ (waiting time)

(iv) Min $\sum_j L_j$ (lateness)

Proof: $\sum_j C_j = \sum_j F_j \boxed{+\sum_j r_j} = \sum_j W_j \boxed{+\sum_j r_j + \sum_j p_j} = \sum_j L_j \boxed{+ \sum_j d_j}$. All squared values are independent of the schedule.