2012-11598
민두기
1-(a).

$$(a_0,\ b_0), (a_1, b_2), (a_2, b_1), (a_3, b_3)$$

1-(b).

$$(a_0,\ b_0), (a_1, b_2), (a_2, b_4), (a_3, b_1), (a_4, b_3)$$

1-(c).

$G_4 = 4$
$G_5 = 2$

2.

There are two chain $(c-1-a-4), (2-b-3-d)$.
If choose $(1, a)$ edge, we cannot choose both $(1, c)\ and\ (4, a)$ and then we cannot make perfect matching. Because c has only one edge, $(1, c)$, and 4 has only one edge $(4, a)$. Similarly, if we choose $(3, b)$ edge, we cannot choose both $(2, b)\ and\ (3, d)$ and then we cannot make perfect matching. Because 2 has only one edge, $(2, b)$, and d has only one edge $(3, d)$. Therefore, we should choose $(1, c)\ or\ (4, a)$ earlier than $(1, a)$, and should choose $(2, b)\ or\ (3, d)$ earlier than $(3, b)$, then we can make perfect matching.

$$\frac{6!}{3! * 3!} * (3! - 2) * (3! - 2) = 320$$

$(3! - 2)$ is number of random chain' s order – number of order $(3, b)$ or $(1, a)$ is first. $\frac{6!}{3!*3!}$ is random edge order / two chain' s order.

Total **320** orders give us a perfect matching.

Simple test is that just check

1. $(3, b)$ is earlier than $(2, b)\ and\ (3, d)$, if then, this order does not give perfect matching.
2. $(1, a)$ is earlier than $(1, c)\ and\ (4, a)$, if then, this order does not give perfect matching.

If pass above two condition, then that order gives perfect matching.

3-(a).

All two query x can make edge always, because all A, B, C can make edge with x. Also two query y can make edge always, because two x make just two edge. That means at least two B or C remain, then they can make edge with two y.

Therefore, the greedy algorithm will assign at least 4 of these 6 queries.

3-(b).

Proper query sequence is xxzzzz. Optimum off-line algorithm assigns 4-queries on this sequence. If greedy algorithm assigns C-x and C-x, then only 2 queries assigned, half of the optimum value.