Maoting Ren
mren

6-1

order the jobs in non-decreasing order of $w^i/(1 – a^{pi})$, and the following is proof.

x	p_i	p_j	y

Graph 1

x	p_j	p_i	y

Graph 2

assume our best solution is graph 1, so when we exchange the position of p_i and p_j, it should be satisfy:

$$w_i * (1 – a^{-pi-x}) + w_j * (1 – a^{-pj-x-pi}) <= w_j * (1 – a^{-pj-x}) + w_i * (1 – a^{-pi-x-pj})$$
$$=> \quad w_i * (a^{-pi-x-pj} - a^{-pi-x}) <= w^j * (a^{-pj-x-pi} - a^{-pj-x})$$
$$=> \quad w^i * (1 – a^{pj}) <= w^j * (1 – a^{pi})$$
$$=> \quad w^i / (1 – a^{pi}) <= w^j / (1 – a^{pj})$$

so we can know that the best solution should order the jobs in non-decreasing order of $w^i / (1 – a^{pi})$.

6-2

Firstly sort the girls and boys by height in non-increasing order, then traverse each girl. And for each girl, add the boys, whose height bigger than that girl, into a binary search tree by weight as the key. Then find the first boy whose weight larger than that girl from the binary search tree, and delete that boy from BST, that means we find one pair, if we can't find such a boy whose weight is larger than that girl, that means we can't find a boy match that girl. Then we compute the next girl, and add the left boys, whose height bigger than than girl, into the binary search tree, and do the same thing until the last girl.

```
findMatch(vector<pair<int, int>> girls, vector<pair<int, int>> boys)
    int sum = 0, num = 0;
    sort girls and boys in non-increasing order by height;
    BST bst; // this is a binary search tree and key is weight
    for( int  i=0; i< girls.size(); i++)
        for(int j = num; j < boys.size(); j++)
            if(boys[j].first >= girls[i].first) // if this boy's height bigger than this girl
                bst.insert(boys[j]); // insert this boy into the binary search tree
                num++;
        find the first boy whose weight is larger than this girl from bst
        if can find such a node from bst
            sum++;
            delete this boy from the bst;
    return sum;
```

6-3

6-4