# Car Fleet

```
class Solution {
public:
    int carFleet(int target, vector<int>& position, vector<int>& speed) {
        int i;
        unordered_map<int,float> map;
        unordered_map<int,stack<int>> map1;

        int curleader;

        for (i = 0;i<speed.size();i++){
          map[position[i]] = calculate_time(position[i],speed[i],target);
        }

        sort(position.begin(), position.end());

        curleader = position[position.size()-1];

        for (i = position.size() -1;i>=0;i--){
          if ((map[curleader] - map[position[i]]) >= 0 ){
            map1[curleader].push(position[i]);
          }else{
            curleader = position[i];
            map1[curleader].push(position[i]);
          }
        }

        return map1.size();

    }

    float calculate_time(int pos,int speed,int target){
      return (float(target)-float(pos))/float(speed);
    }
```

```
};
```