

Solution

Analysis by Jesse Choe

Binary search on the answer for X . We can check whether the sum of the distance of all cows is at least M miles for finding X .

We know that a given cow i travels a total of d_i **miles** in $t_i \times d_i + w_i$ **minutes**. Thus, we know that a given cow i travels a "full" distance of $\frac{x}{(t_i \times d_i + w_i)} \times d_i$ miles. Then, we know that cow i travels a "partial" distance of $\frac{rem}{t_i}$, where rem is $\min(X \pmod{t_i \times d_i + w_i}, t_i \times d_i)$. Then, we can check if the sum of the "full" distance and the "partial" distance for all cows i is at least M miles.

For the second task of the problem: finding the total distance covered by each cow, we can subtract the sum of the "full" distance and "partial" distance to the current distance that remains in the race, initialized at M , so that all cows finish the race in conclusion of meeting the requirement.

My C++ code:

```
#include <bits/stdc++.h>

using namespace std;
using ll = long long;

struct Cow {
    int t, d, w;
};

int n, m;
vector<Cow> arr;

bool works(ll time){
    ll res = 0;
    for(int i=0; i<n; i++){
        ll intervals = (arr[i].t * arr[i].d + arr[i].w);
        ll totIntervals = (time / intervals);
        ll rem = (time % intervals);
        if(rem >= arr[i].t * arr[i].d){
```

```

        rem = arr[i].t * arr[i].d;
    }
    ll balloons = (totIntervals * arr[i].d) + (rem / arr[i].t);
    res += balloons;
}
// Check if the total distance traveled by all cows is at least M
return res >= m;
}

void balloons(ll time){
    ll cur = m;
    for(int i=0; i<n; i++){
        ll intervals = (arr[i].t * arr[i].d + arr[i].w);
        ll totIntervals = (time / intervals);
        ll rem = (time % intervals);
        if(rem >= arr[i].t * arr[i].d){
            rem = arr[i].t * arr[i].d;
        }
        ll balloons = (totIntervals * arr[i].d) + (rem / arr[i].t);
        if(balloons >= cur){
            balloons = cur;
        }
        // Subtract the total balloons from the current running total.
        cur -= balloons;
        // Print the balloons
        cout << balloons << " ";
    }
    cout << endl;
}

int main(){
    cin >> m >> n;
    for(int i=0; i<n; i++){
        int t, d, w; cin >> t >> d >> w;
        arr.push_back({t, d, w});
    }
    ll l = 0, r, ans;
    r = 1;
    while(!works(r)){
        r *= 2;
    }
    ans = r;
    while(l <= r){
        ll mid = l + (r-l) / 2;
        if(works(mid)){
            ans = mid;
        }
    }
}

```

```
        r = m-1;
    } else {
        l = m+1;
    }
}
cout << ans << endl;
balloons(ans);
}
```