telecab **Autor: Constantin Gălăţan**

Soluţia 1. 100 puncte (Constantin Gălăţan)

Cerinţa 1).
Traseul telecabinei este unic determinat. Fie `h[1], h[2], ..., h[n]` şirul înălţimilor cotelor şi `p[1], p[2], …, p[n]` un şir având semnificaţia: `p[i]` = poziţia primei cote mai înaltă decât cota i.
Exemplu:
pentru h = {**4, 5, 2, 2, 1, 3, 5, 3, 3}, p = {2, 0, 6, 6, 6, 7, 8, 0, 0}**
Se observă că dacă `p[i] = j`, atunci, `p[i+1] = j, p[i+2] = j, …, p[j-1] = j`. Telecabina uneşte în mod direct cotele i şi `j`. Pentru o soluţie liniară, se menţine o stivă în care se introduc pe rând toate cotele. Dacă `i` e cota din vârful stivei şi `j` este cota curentă, atunci înainte de a introduce `j` în stivă, cât timp `h[j] > h[i]`, se marchează `p[i] = j` şi se scoate i din stivă. Complexitate: O(n).

Cerinţa 2)
Pentru o soluţie de tip greedy se acordă 70% din punctajul acordat pentru această cerinţă.
Soluţia oficială, care primeşte punctajul maxim, foloseşte recursie cu memorizare după cum urmează: definim o funcţie recursivă **Timp(i, s)** – timpul minim necesar să se atingă o cotă cu numărul de ordine cel puţin `k` (este posibil ca telecabina să treacă *pe deasupra* cotei `k`), pornind de la cota `i` cu suma `s`.

Funcţia **Timp(i, s)** se defineşte recurent astfel:

$$\begin{cases} 0 \text{ dacă } i \geq k \\ 1 + \mathrm{Timp}(i+1, s) - \text{segment orizontal} \\ \mathrm{dist}(i, j) + \mathrm{Timp}(j, s - (h[j] - h[i])) - \text{urcare pe segmentul } [i\ j] \\ \min(1 + \mathrm{Timp}(i+1, s - (h[i] - h[i+1])), \mathrm{dist}(i, i+1) + \mathrm{Timp}(i+1, s)) \text{ - coborâre pe segmentul } [i, i+1] \end{cases}$$

Întrucât există două opţiuni de coborâre, pentru a se evita recalcularea timpului minim pentru stările cu aceeaşi sumă `s` şi cotă `i`, se reţin timpii într-un tablou `tmin[i][s]` (memoizare).
Complexitatea soluţiei: O(n + K * S).

Soluţia 2 – Marius Dumitran

Subpunctul 2.
Rezolvare cu programare dinamica:
Vom folosi o matrice M[i][j] unde M[i][j] este cea mai scurta distanta cu care putem ajunge la cota i, folosind j bani. M[i][j] = INF daca nu putem ajunge in punctul i folosind j bani

De asemenea ne vom ajuta si de vectorii Next, unde Next[i] este cota urmatoare lui i in traseul telecabinei, si H[i] unde H[i] este inaltimea punctului i.

Initializare : M[1][0] = 0.

Rezolvare :
```
for( int i = 1; i <= K; ++i)
        for( int j = 0; j <= S; ++j)
        {
                // urcare
                if( h[ i ] < h[ next[ i ]] )
                        if( M[ Next[ i ]][ j + h[ j ] - h[ i ] ] > M[ i ][ j ] + dist( i, next[ i ])
                                M[ Next[ i ]][ j + h[ j ] - h[ i ] ] = M[ i ][ j ] + dist( i, next[ i ];
```

```
            if( h[ i ] > h[ next[ i ]] )
            {
                    // coborare
                    if( M[ i + 1 ][ j ] > M[ i ][ j ] + dist( i , next[ i ]) // fara bani
                            M[ i + 1 ][ j ] = M[ i ][ j ] + dist( i , next[ i ]
                    if( M[ i + 1][ j + h[ i ] - h [ j ]] > M[ i ][ j ] + 1 )
                            M[ i + 1][ j + h[ i ] - h[ j ] = M[ i ] [ j ]  + 1;
            }
    }
```

Complexitate O (K * S).

Soluţia 3 (prof. Stelian Ciurea)
Cerinta 1:

I - construiesc la citire 10 vectori: d_1,d_2,...,d_{10}; in di retin pozitiile unde apar cote de inaltime i. din construictie acesti vectori vor fi sortati.
II - trebuie sa determin cotele care formeaza traseul telecabinei. pentru aceata parcurg cotele si pentru cota din pozitia i avand inaltimea h[i], determin prin cautare in vectorii $d_{h[i]+1}$,$d_{h[i]+2}$,,,d_{10} cea mai mica valoare mai mare decat i - aceasta va reprezenta pozitia primei cote mai mari decat cea din pozitia i. din enunt, traseul telecabinei va fi de la cota i
la cota determinata astfel. vectorii fiind sortati, cautarea o voi face binar. retin astfel pozitiile cotelor pe unde va trece traseul telecabinei.
cunoscand traseul telecabinei determinarea sumei distantelor este imediata.

Complexitate: O(n*10*log(n))

Cerinta 2:
I - deoarece pentru portiunile in care telecabina urca sau merge pe orizontala nu am de ales, determin pentru acestea durata si suma necesara.
II - retin in doi vectori distantele si diferentele de inaltime pentru portiunile unde telecabina coboara.
III - sortez descrescator cei doi vectori dupa valorile din vectorul de distante. ca sa minimzez timpul de coborare, prefer sa-mi cheltuiesc suma ramasa pentru a parcurge cat mai repede portiunile cele mai lungi! eventuala suma ramasa (in cazul in care nu imi ajung banii pentru ultima distanta astfel determinata), o cheltuiesc pentru a cobora rapid portiuni cu diferente de inaltime mici (deoarece inaltimile sunt dintr-un interval foarte mic, probabilitatea sa gasesc diferente de inaltime egale cu 1 pe care sa le pot parcurge cu suma ramasa este foarte mare).

Complexitatea este cea a sortarii: O(n*log(n))