

Feladat

Egy hegyoldal hegycsúcs felé vezető ösvénye mentén egyenlő távolságonként megmértük a terep tengerszint feletti magasságát, és a mért értékeket egy tömbben (vektorban) tároljuk.

(Megfigyeltük, hogy ezek az értékek egyszer sem csökkentek az előző értékhez képest.) Igaz-e, hogy mindig növekedtek?

Megjegyzések

Az, hogy az értékek egyszer sem csökkentek, programozás szempontjából elhagyható információ, de azért a teljesség kedvéért az előfeltételbe beleírtam.

A "mindig növekedtek" utófeltételt kétféleképpen lehet specifikálni:

1. Visszanézés: A $[2..n]$ intervallumon: $v[i - 1] < v[i]$
2. Vagy előrenézés: Az $[1..n - 1]$ intervallumon: $v[i] < v[i + 1]$

Én a visszánézést választom. (Az "egyszer sem csökkentek" előfeltételt hasonlóan visszánézéssel ellenőrzöm az előfeltételben, de a programban nem látom fontosnak ellenőrizni.)

Specifikáció

→Optimista lineáris keresés

$$A = (v: \mathbb{R}^n, l: \mathbb{L})$$

$$Ef = (v = v' \wedge \forall i \in [2..n]: v[i - 1] \leq v[i])$$

$$Uf = (Ef \wedge l = \forall i \in [2..n]: v[i - 1] < v[i])$$

$$l = \forall i \in [2..n]: v[i - 1] < v[i]$$

↔

$$\neg l = \exists i \in [2..n]: v[i - 1] \geq v[i]$$

Algoritmus

A feladatot a lineáris keresés programozási tételének "optimista" változatára vezetjük vissza.

Mivel a tétel kimenetei között szereplő index a feladat szempontjából nem érdekes, ezért az $ind := i$ értékadás elhagyható az algoritmusból.

$$\beta(i) \sim v[i - 1] < v[i]$$

$m..n \sim 1..n$

$l \sim \neg l$

$l, i = igaz, 2$
$l \wedge i \leq n$
$l := v[i-1] \geq v[i]$
$i := i + 1$

Implementáció

Adattípusok megvalósítása

Kódozaskor a v vektort $vector < int >$ v -ként deklaráljuk, amelynek mérete $v.size()$ alakban érhető el.

Mivel a vektorok mérete nincs a vektorban eltárolva, minden $v.size()$ híváskor újra ki kellene azt számolni. Ennek elkerülése végett bevezeték egy end nevű segédváltozót, amit a ciklusfeltételben fogok használni a $v.size()$ helyett.

```
int end = v.size()
```

C++

C++-ban a vektor 0-tól indexelődik, ezért a tervbeli ciklus nem a $[2..n]$, hanem az $[1..n-1]$ intervallumot, pontosabban az $[1..end)$ jobb oldalon nyílt intervallumot futja be.

A C++ programozási nyelv egy hasznos tulajdonsága miatt, miszerint a függvények visszatérési értékkel bármikor kiléphetnek, elhagyható az l logikai változó is. Helyette a $return true/false$ utasítást használom.

Struktogram:

$i = 1$
$i < end$
$v[i-1] \geq v[i]$
$return false$ -
$i := i + 1$
$return true$

Egyszerűbb for ciklussal:

$i = 1..end$
$v[i-1] \geq v[i]$
$return false$ -
$return true$

Így az utófeltétel funkciójának forráskódja a következő lesz:

```
bool increasing(vector < int > &v){  
    int end = v.size();  
    for(int i = 0; i < end; i++){
```

```

        if (v[i - 1] >= v[i]){
            return false;
        }
    }
    return true;
}

```

Bemenő adatok formája

A bemenő adatokat egy szöveges állományból kell a vektorba beolvasni. Az állományban a megadott értékeket szóközzel, tabulátor jelekkel, sorvége jelekkel, vagy egyéb kompatibilis módon elválasztva kell beírni.

Például

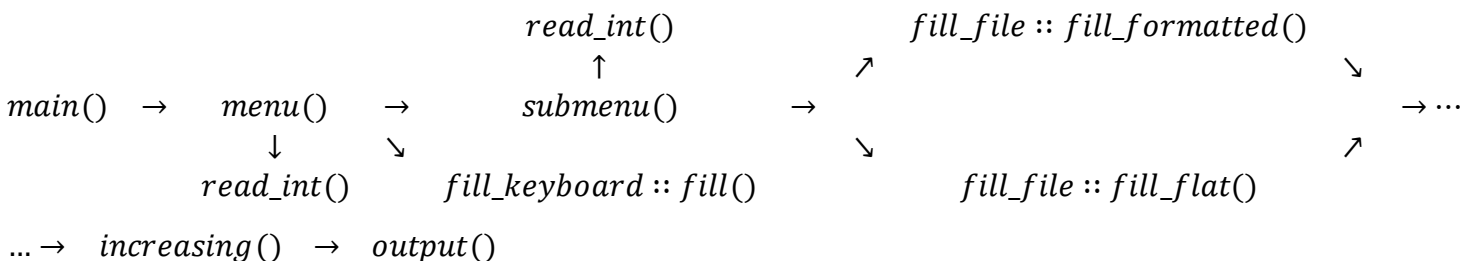
1 2 3 4 5

1 → 2 → 3 → 4 → 5

1
2
3
4
5

Függvények kapcsolódási szerkezete

A fő funkciók így hívják egymást:



Igen, a feladat részekre darabolását alaposan túlzásba vittem...

Tesztelési terv

A feladat specifikációjára épülő (fekete doboz) tesztesetek:

Érvényes tesztesetek:

(érvénytelen tesztesetek nincsenek)

A. Intervallum tesztesetei:

1. Egyetlen bemenő adat sincs (üres vagy csak egy elválasztójelet tartalmazó fájl esete)
 - i. (a1i.txt: [] – válasz: igaz)
 - ii. (a1ii.txt: [] – válasz: igaz)

2. Egy érték szerepel csak
 - i. (a2.txt: [1] – válasz: igaz)
3. Több csökkenő érték, és a végén egy növő.
 - i. (a3.txt: [5, 4, 3, 2, 1, 2] – válasz: hamis)
4. Több növő érték, és a végén egy csökkenő.
 - i. (a4.txt: [1, 2, 3, 4, 5, 4] – válasz: hamis)
5. Több csökkenő érték, és a végén egy nem növő.
 - i. (a5.txt: [5, 4, 3, 2, 1, 1] – válasz: hamis)
6. Több növő érték, és a végén egy nem növő.
 - i. (a6.txt: [1, 2, 3, 4, 5, 5] – válasz: hamis)
7. Több csökkenő érték, és az elején egy növő.
 - i. (a7.txt: [4, 5, 4, 3, 2, 1] – válasz: hamis)
8. Több növő érték, és az elején egy csökkenő.
 - i. (a8.txt: [2, 1, 2, 3, 4, 5] – válasz: hamis)
9. Több csökkenő érték, és az elején egy nem növő.
 - i. (a9.txt: [5, 5, 4, 3, 2, 1] – válasz: hamis)
10. Több növő érték, és az elején egy nem növő.
 - i. (a10.txt: [1, 1, 2, 3, 4, 5] – válasz: hamis)

B. Lineáris keresés tesztesetei:

1. Egyetlen érték szerepel csak
 - i. (b1.txt: [1] – válasz: igaz)
2. Növő sorozat
 - i. (b2.txt: [1, 2, 3] – válasz: igaz)
3. Csökkenő sorozat
 - i. (b3.txt: [3, 2, 1] – válasz: hamis)
4. Növő sorozat közepén egy nem növő elemmel
 - i. (b4.txt: [1, 2, 3, 3, 4, 5] – válasz: hamis)
5. Növő sorozat közepén egy csökkenő elemmel
 - i. (b5.txt: [1, 2, 3, 2, 4, 5] – válasz: hamis)

C. Különleges értékek esetei (Billentyűzetről tesztelendő):

1. Üres sztring, mint érték (Figyelmen kívül hagyja, tovább vár a bemenetre, mintha misem történt volna.)
2. Negatív számok (Az elvárt módon működik.)
3. Betűt tartalmazó értékek esete (Figyelmen kívül hagyja a betűt.)
4. Speciális karakterek esete (Hibaüzenetet ír ki)

A megoldó programra épülő (fehér doboz) tesztesetek:

1. Hibás vagy nem létező állománynév megadása. (Hasznos hibaüzenetet ír ki.)
2. Paraméterként megadott fájlnev. (Az elvárt módon működik.)
3. Ismételt futtatás kipróbálása. (Az elvárt módon működik.)
4. Olyan állomány olvasása, ahol egy sorban több érték is található egyetlen illetve több szóközzel és/vagy tabulátor jellel elválasztva (d1.txt).
5. Olyan állomány olvasása, ahol minden érték külön sorban van (d2.txt).
6. Olyan állomány olvasása, ahol az utolsó sort nem zárja sorvége jel (d3.txt).
7. Főprogram ciklusának ellenőrzése: olyan bemenő adatokkal, amelyekre a ciklus:
 - a. egyszer sem fut le (d4.txt)
 - b. pontosan egyszer fut le (d5.txt)
 - c. többször lefut, igaz logikai értékkel lép ki (d6.txt)
 - d. vagy hamis logikai értékkel lép ki (d7.txt).
8. Stressz-teszt, 5000-5000 értékkel, majd 1.000.000-1.000.000 értékkel (! *stress – test.bat*)

Minden teszteset (kivéve stressz teszt) lefuttatása egyszerre: !*test.bat*