

PROGRAMAZIOAREN METODOLOGIA

Kudeaketaren eta Informazio Sistemen Informatikaren Ingeniaritzako Gradua
Bilboko Ingeniaritza Eskola (UPV/EHU)
Lengoaia eta Sistema Informatikoak Saila

1. maila

2019-2020 ikasturtea

31 taldea

Entregatu beharreko lana

6. gaia – Programa errekurtsiboen eraldaketa

2 puntu

2020ko apirilaren 20a

1. ARIKETA (*Burstall-en metodoa*) – (2 puntu)

Har dezagun *edk* izeneko funtzioa. Funtzio horrek sarrerako datu gisa, negatiboak ez diren zenbaki osoz eratutako *s* zerrenda eta negatiboak ez diren *n* eta *x* bi zenbaki oso izango ditu. Emaizta gisa, *s* zerrendako eskuineko ertzetik hasi eta *x*-ren desberdinak diren *n* elementu kenduz geldituko den zerrenda itzuliko du. Sarrera gisa emandako *n* balioa *s* zerrendatik kendu daitezkeen elementu kopurua baino handiagoa baldin bada, emaitza gisa itzuliko den zerrenda eraikitzeko, ahal diren elementuak kendu egin beharko dira *s*-tik eta emaitza izango den zerrendako ezkerreko ertzean beste zenbat elementu kendu beharko liratekeen —baina ezin diren kendu, ez daudelako— adierazten duen balioaren negatiboa ipini beharko da.

edk funtzioaren definizio formala Haskell lengoaiaren honako hau da:

```
edk :: ([Int], Int, Int) -> [Int]
{- Hasierako baldintza: ez_negatiboa(s) ∧ n ≥ 0 ∧ x ≥ 0 -}
edk(s, n, x)
| n == 0                = s                                (# 1)
| hutsa_da(s)           = [- n]                            (# 2)
| azkena(s) == x        = edk(azkena_kendu(s), n, x) ++ [x] (# 3)
| azkena(s) /= x        = edk(azkena_kendu(s), n - 1, x)  (# 4)
```

edk funtzioaren definizio horretan:

- *ez_negatiboa* izeneko predikatua, zenbaki osozko *ℓ* zerrenda bat emanda, *ℓ* zerrendako elementu guztiak ez-negatiboak baldin badira, hau da, zero baino handiagoak edo berdinak baldin badira, True eta bestela, False itzuliko duen predikatua da. Predikatu horren definizio formala honako hau da:

$$ez_negatiboa(\ell) \equiv \forall k(1 \leq k \leq \text{luzera}(\ell) \rightarrow \text{elem_pos}(k, \ell) \geq 0)$$
- *luzera* izeneko funtzioa, zerrenda bat emanda, zerrendako elementu kopurua itzuliko duen funtzioa da.
- *elem_pos* izeneko funtzioa, posizio egoki bat eta zerrenda bat emanda, zerrendako posizio horretan dagoen elementua itzuliko duen funtzioa da. Posizioak ezkerretik eskuinera zenbatzen dira 1etik hasita. Posizioa egokia ez bada, funtzio honek errore-mezua aurkeztuko du.
- *hutsa_da* izeneko funtzioa, zerrenda bat emanda, zerrenda hutsa baldin bada, True eta bestela, False itzuliko duen funtzioa da.
- *azkena* izeneko funtzioa, zerrenda bat emanda, zerrendako azkeneko elementua (eskuineko ertzekoa) itzuliko duen funtzioa da. Sarrerako zerrenda hutsa baldin bada, funtzio honek errore-mezua aurkeztuko du.
- *azkena_kendu* izeneko funtzioa, zerrenda bat emanda, zerrendako azkeneko elementua (eskuineko ertzekoa) kenduz gelditzen den zerrenda itzuliko duen funtzioa da. Sarrerako zerrenda hutsa baldin bada, funtzio honek errore-mezua aurkeztuko du.

Adibideak *edk* funtzioarentzat:

- $edk([9, 0, 7, 3, 8, 3, 3], 2, 3) = [9, 0, 3, 3, 3]$
3ren desberdinak diren 2 elementu kendu dira eskuineko ertzetik hasita.
- $edk([9, 0, 7, 3, 8, 3, 3], 2, 0) = [9, 0, 7, 3, 8]$
0ren desberdinak diren 2 elementu kendu dira eskuineko ertzetik hasita.
- $edk([9, 7, 3, 8], 10, 5) = [-6]$
5en desberdinak diren 10 elementu kendu nahi dira eskuineko ertzetik hasita, baina bakarrik 4 elementu kendu daitezke. Beraz, lau elementu horiek kendu dira eta emaitza izango den zerrendaren hasieran (ezkerreko ertzean) beste zenbat elementu kendu beharko liratekeen —baina ezin diren kendu, ez daudelako— adierazten duen balioaren negatiboa ipini da.

- $\text{edk}([5, 5, 5, 5], 3, 5) = [-3, 5, 5, 5, 5]$
 Sen desberdinak diren 3 elementu kendu nahi dira eskuineko ertzetik hasita, baina ez dago kendu daitekeen elementurik, zerrendan 5ekoak bakarrik baitaude. Beraz, ez da elementurik kendu eta emaitza izango den zerrendaren hasieran (ezkerreko ertzean) beste zenbat elementu kendu beharko liratekeen —baina ezin diren kendu, ez daudelako— adierazten duen balioaren negatiboa ipini da.
- $\text{edk}([5, 5, 8, 7, 5], 3, 5) = [-1, 5, 5, 5, 5] \leftarrow \text{Adibide hau erabili inbariantea kalkulatzeko}$
 Sen desberdinak diren 3 elementu kendu nahi dira eskuineko ertzetik hasita, baina bakarrik 2 elementu kendu daitezke. Beraz, bi elementu horiek kendu dira eta emaitza izango den zerrendaren hasieran (ezkerreko ertzean) beste zenbat elementu kendu beharko liratekeen —baina ezin diren kendu, ez daudelako— adierazten duen balioaren negatiboa ipini da.

Burstall-en metodoa erabiliz, *edk* funtzioa eraldatu eta gauza bera egiten duen funtzio iteratibo bat lortu.

Puntuazioa	
a) Hasierako itzulpena – 0,025 puntu	e) Baldintza eta bukaera – 0,250 + 0,250 puntu
b) Errekurrentzia-erlazioa – 0,025 puntu	f) While-eko aginduak – 0,775 puntu
c) Inbariantea – 0,575 puntu	g) Programa osoa idaztea – 0,050 puntu
d) Hasieraketa – 0,050 puntu	

OHARRA:

Ariketa gainditzeko c, d, e eta f ataletan, atal horietako bakoitzari dagokion puntuazioaren erdia lortu beharko da gutxienez.