

# KONPUTAGAILU EREDU ABSTRAKTUAK: SUBSET TO SAT

eman ta zabal zazu



Universidad  
del País Vasco

Euskal Herriko  
Unibertsitatea

Iñigo Gastesi Arribas

<b>Sarrera</b>	<b>3</b>
<b>Egindakoa</b>	<b>3</b>
<b>Optimizazioak</b>	<b>4</b>
<b>Arazoak</b>	<b>5</b>
<b>Exekuzioa</b>	<b>5</b>

## 1. Sarrera

Subset-sum problema bat izanik, problema hau sat problema batera bihurtu behar dugu. Zenbaki multzo bat dago eta zenbaki bat zenbaki multzo honen kanpoan, subset-sum problematik, begiratu behar dugu ea dagoen konbinaketa bat zenbaki multzoko zenbakiekin non, beraien baturaren balioa multzotik kanpo dagoen zenbakiaren balioaren berdina den. Aldiz, SAT problema "formula" bat daukagu, formula hau disjuntzioen konjuntzioekin osatuta dago. Konjuntzio multzo bakoitzean hainbat aldagai daude, aldagai guztiei True edo False balioa eman behar zaie formula betetzen duen aldagaien balioen esleipena aurkitu arte. Gure kasuan subset-sum problematik, sat problemara pasa behar dugu, horretarako zenbaki multzoa eta batuketa, kodetu behar ditugu SATeko dalgaiak eta klausulak bezala.

## 2. Egindakoa

Hasteko kodeketetarekin hasi naiz, baina nire kasuan, kodeketa egindakoan egiten diren taulak, eskubitik ezkerreko ordenatu beharrean, alderantziz egin dut, ezkerretik eskubira, hau for bat iteratzerakoan sinpleagoa egiteko eta geroago, batuketen eta bururako kodeketan hau gauza asko errazten ditu.

Nire programan, matrix izeneko aldagai bat dago, aldagai hau exekuzioan zehar, aldaketak izango ditu eta prematrix beste aldagai bat da non, matrix aldagaia aldatu baino lehen honen berdina izango den.

Hauek izango ziren matrix izandako aldaketak azalpeneko dauden adibideak hartuz.

	-7	6	5	< - - 1
	10	-9	-8	< - - 2
	13	12	-11	< - - 3
17	-16	15	-14	< - - 4
? + 3	? + 2	? + 1	-?	

	20	19	18	< - - 1
	23	22	21	< - - 2
	26	25	24	< - - 3
30	29	28	27	< - - 4
? + 3	? + 2	? + 1	-?	

53 < - > 57	37 < - > 40	33 < - > 20	32 < - > 19	31 < - > 18	
		33	32	31	< - - 1
		36 40	35 39	34 38	< - - 2
	44 48	43 47	42 46	41 45	< - - 3
	52 57	51 56	50 55	49 54	< - - 4
-53	52	51	50	-49	

Azkenean, matrix azkeneko taulan dauden aldagaiak izango zituen, baina bakarrik ezkerreko balioak izango ditu, eskubiko balioak bururakoMatrix izeneko aldagai batean

egongo dira. Bukatzeko prematrix bigarren taularen balioak izango ditu. Lehenengo bi tablak sortzerakoan, klausulak sortzen joaten dira, lehenengo balioak hasieratzeko. Ondoren, balioa bakoitza ea konbinaketan egongo den edo ez.

Hau eta gero batuketan eta bururakoen kodeketa egin behar da. Hau errazteko, hiru matrizeak zeroz bete ditut, matrize karratu bat sortu harte. Adibidez, batuketa bat egiterakoan, goiko balioa dagoen edo ez dagoen jakiteko bakarrik 0 bat dagoen begiratu behar dut.

```
> 0: [85, 86, 87, 88, 0, 0]
> 1: [89, 90, 91, 92, 93, 0]
> 2: [99, 100, 101, 102, 103, 104]
> 3: [111, 112, 113, 114, 115, 116]
> 4: [123, 124, 125, 126, 127, 128]
> 5: [135, 136, 137, 138, 139, 140]
> 6: [147, 148, 149, 150, 151, 152]
> 7: [159, 160, 161, 162, 163, 164]
```

Honengatik, matrizeak ezkerretik eskubira ordenatu ditut, bestela hau egitea ezinezkoa izango zen, eta luzeerekin ibili beharko nintzen kalkulu hauek egiteko. Beraz, hau esanda batuketan eta bururakoen kalkuluak azalduko ditut. Bost kasu daude, lehenengo ilara berezia dena, eskubiko zutabea, zutabe hau ez ditu bururakorik izango batuketa egiterakoan, eskubiko bururakoa balioa du, goiko batuketa balioa du, baina posizioa hori balioen matrizean ez du aldagaiarik, goiko batuketa ez du baliorik eta balioen matrizean balioa du eta aldagai guztiak balioa dute. Kasu bakoitzaren arabera elkartrukaketa bat, 2x2 edo 4x4 batuketa eta bururakoa kalkulatu da.

Bukatzeko azken ilararen kalkulua egiten da eta emaitza string bezala gordetzen da.

All subset ere inplementatuta dago, all subset en emaitza begiratzeko zenb.txt fitxategia begiratu behar da eta azkeneko klausuletan emaitza guztiak ezeztatuak egongo dira.

### 3. Optimizazioak

Optimizazio bakar bat inplementatuta dago. Optimizazio honetan, sortutako tabla emaitza bitarrean daukan luzeraren zabalera izango du gehienez eta azkeneko zutabeko bururakoen balioak 0 izango da. Horrela aldagai asko aurrezten dira. Optimizazio hau asko nabaritzen da zenbaki asko errepikatzen direnean

## **4. Arazoak**

Exekuzio denborekin arazo handi bat dut, 2119841 zenbakitik aurrera, exekuzio denborak asko irauten dute, zenbaki honek 12 ordu behar ditu eta hurrengo zenbakia 24 ordu ondoren oraindik ebazten zegoen. Kissat-ekin probak egin nituen eta kissat 30 minutu eta 1 ordu eta 30 minutu behar izan zituen.

## **5. Exekuzioa**

Programa exekutatzeko bakarrik testSubsetSum fitxategia exekutatu behar da, ez dago ezer berezirik, ala ere nire\_proba.py fitxategia dago, Subset-sum bakar bat exekutatu nahi baduzu