Programa 2

Algoritmos de aproximación

Información del curso

Complejidad Computacional - Facultad de Ciencias, UNAM.

- Profesor: María de Luz Gasca Soto
- Ayudante José Luis Vázquez Lázaro
- Ayudante Jorge Luis García Flores

Descripción de la práctica

En el archivo de especificación de la prática viene todo explicado `doc.pdf`.

Entorno

- `Python`: Python 3.7.4

Ejecución del programa

Se tiene un archivo `Makefile` para la ejecución y limpieza del programa.

Para ejecutar **Subset-sum**:

```
$ make run_subset
```

Para ejecutar **Bin Packing**:

```
$ make run_bin
```

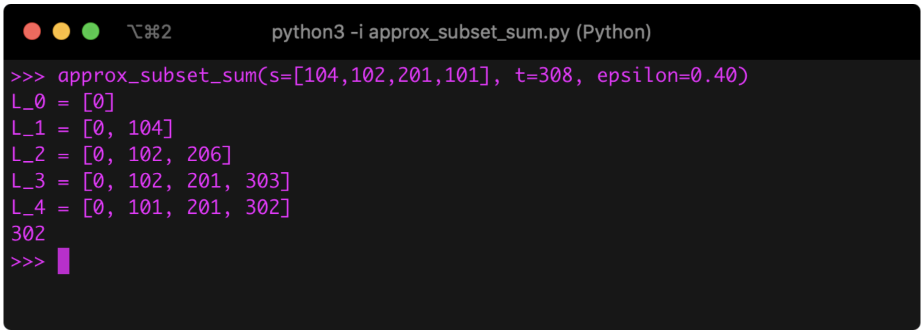
Para limpiar:

```
$ make clean
```

Ejecuciones

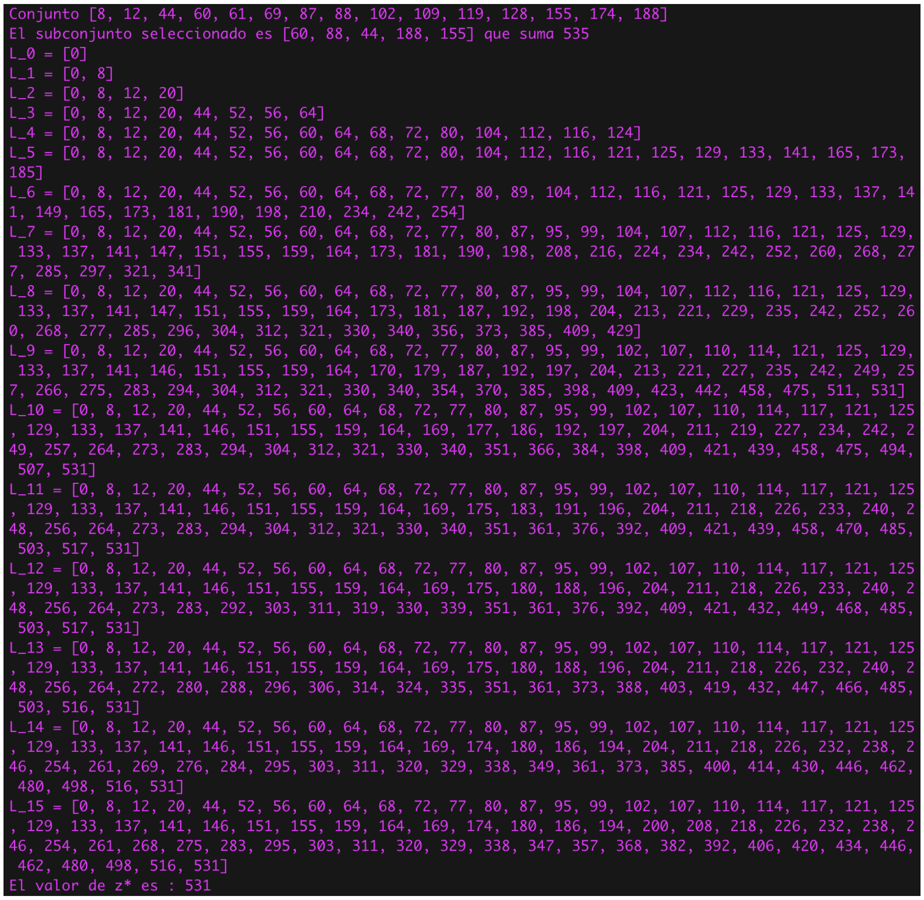
Subset-sum

Parámetros tomados de Introduction to Algorithms de Thomas H. Cormen, en la página 1131.



```
>>> approx_subset_sum(s=[104,102,201,101], t=308, epsilon=0.40)
L_0 = [0]
L_1 = [0, 104]
L_2 = [0, 102, 206]
L_3 = [0, 102, 201, 303]
L_4 = [0, 101, 201, 302]
302
>>> 
```

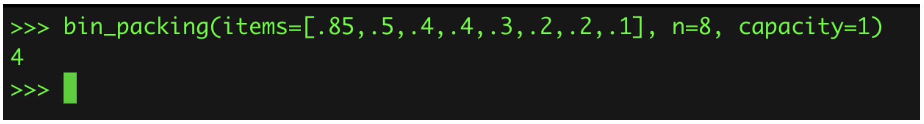
Ejecución arbitraria mostrando su aproximación



```
Conjunto [8, 12, 44, 60, 61, 69, 87, 88, 102, 109, 119, 128, 155, 174, 188]
El subconjunto seleccionado es [60, 88, 44, 188, 155] que suma 535
L_0 = [0]
L_1 = [0, 8]
L_2 = [0, 8, 12, 20]
L_3 = [0, 8, 12, 20, 44, 52, 56, 64]
L_4 = [0, 8, 12, 20, 44, 52, 56, 60, 64, 68, 72, 80, 104, 112, 116, 124]
L_5 = [0, 8, 12, 20, 44, 52, 56, 60, 64, 68, 72, 80, 104, 112, 116, 121, 125, 129, 133, 141, 165, 173,
185]
L_6 = [0, 8, 12, 20, 44, 52, 56, 60, 64, 68, 72, 77, 80, 89, 104, 112, 116, 121, 125, 129, 133, 137, 14
1, 149, 165, 173, 181, 190, 198, 210, 234, 242, 254]
L_7 = [0, 8, 12, 20, 44, 52, 56, 60, 64, 68, 72, 77, 80, 87, 95, 99, 104, 107, 112, 116, 121, 125, 129,
 133, 137, 141, 147, 151, 155, 159, 164, 173, 181, 190, 198, 208, 216, 224, 234, 242, 252, 260, 268, 27
7, 285, 297, 321, 341]
L_8 = [0, 8, 12, 20, 44, 52, 56, 60, 64, 68, 72, 77, 80, 87, 95, 99, 104, 107, 112, 116, 121, 125, 129,
 133, 137, 141, 147, 151, 155, 159, 164, 173, 181, 187, 192, 198, 204, 213, 221, 229, 235, 242, 252, 26
0, 268, 277, 285, 296, 304, 312, 321, 330, 340, 356, 373, 385, 409, 429]
L_9 = [0, 8, 12, 20, 44, 52, 56, 60, 64, 68, 72, 77, 80, 87, 95, 99, 102, 107, 110, 114, 121, 125, 129,
 133, 137, 141, 146, 151, 155, 159, 164, 170, 179, 187, 192, 197, 204, 213, 221, 227, 235, 242, 249, 25
7, 266, 275, 283, 294, 304, 312, 321, 330, 340, 354, 370, 385, 398, 409, 423, 442, 458, 475, 511, 531]
L_10 = [0, 8, 12, 20, 44, 52, 56, 60, 64, 68, 72, 77, 80, 87, 95, 99, 102, 107, 110, 114, 117, 121, 125
, 129, 133, 137, 141, 146, 151, 155, 159, 164, 169, 177, 186, 192, 197, 204, 211, 219, 227, 234, 242, 2
49, 257, 264, 273, 283, 294, 304, 312, 321, 330, 340, 351, 366, 384, 398, 409, 421, 439, 458, 475, 494,
 507, 531]
L_11 = [0, 8, 12, 20, 44, 52, 56, 60, 64, 68, 72, 77, 80, 87, 95, 99, 102, 107, 110, 114, 117, 121, 125
, 129, 133, 137, 141, 146, 151, 155, 159, 164, 169, 175, 183, 191, 196, 204, 211, 218, 226, 233, 240, 2
48, 256, 264, 273, 283, 294, 304, 312, 321, 330, 340, 351, 361, 376, 392, 409, 421, 439, 458, 470, 485,
 503, 517, 531]
L_12 = [0, 8, 12, 20, 44, 52, 56, 60, 64, 68, 72, 77, 80, 87, 95, 99, 102, 107, 110, 114, 117, 121, 125
, 129, 133, 137, 141, 146, 151, 155, 159, 164, 169, 175, 180, 188, 196, 204, 211, 218, 226, 233, 240, 2
48, 256, 264, 273, 283, 292, 303, 311, 319, 330, 339, 351, 361, 376, 392, 409, 421, 432, 449, 468, 485,
 503, 517, 531]
L_13 = [0, 8, 12, 20, 44, 52, 56, 60, 64, 68, 72, 77, 80, 87, 95, 99, 102, 107, 110, 114, 117, 121, 125
, 129, 133, 137, 141, 146, 151, 155, 159, 164, 169, 175, 180, 188, 196, 204, 211, 218, 226, 232, 240, 2
48, 256, 264, 272, 280, 288, 296, 306, 314, 324, 335, 351, 361, 373, 388, 403, 419, 432, 447, 466, 485,
 503, 516, 531]
L_14 = [0, 8, 12, 20, 44, 52, 56, 60, 64, 68, 72, 77, 80, 87, 95, 99, 102, 107, 110, 114, 117, 121, 125
, 129, 133, 137, 141, 146, 151, 155, 159, 164, 169, 174, 180, 186, 194, 204, 211, 218, 226, 232, 238, 2
46, 254, 261, 269, 276, 284, 295, 303, 311, 320, 329, 338, 349, 361, 373, 385, 400, 414, 430, 446, 462,
 480, 498, 516, 531]
L_15 = [0, 8, 12, 20, 44, 52, 56, 60, 64, 68, 72, 77, 80, 87, 95, 99, 102, 107, 110, 114, 117, 121, 125
, 129, 133, 137, 141, 146, 151, 155, 159, 164, 169, 174, 180, 186, 194, 200, 208, 218, 226, 232, 238, 2
46, 254, 261, 268, 275, 283, 295, 303, 311, 320, 329, 338, 347, 357, 368, 382, 392, 406, 420, 434, 446,
 462, 480, 498, 516, 531]
El valor de z* es : 531
```

Bin Packing

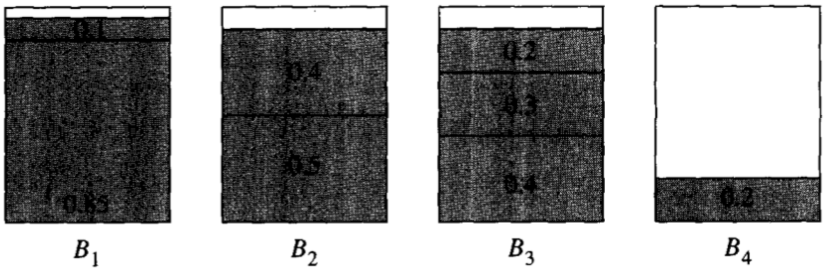
Parámetros tomados de Foundations of Algorithms Using C++ Pseudocode de Richard E. Neapolitan y Kumarss Naimipour, en la página 399.



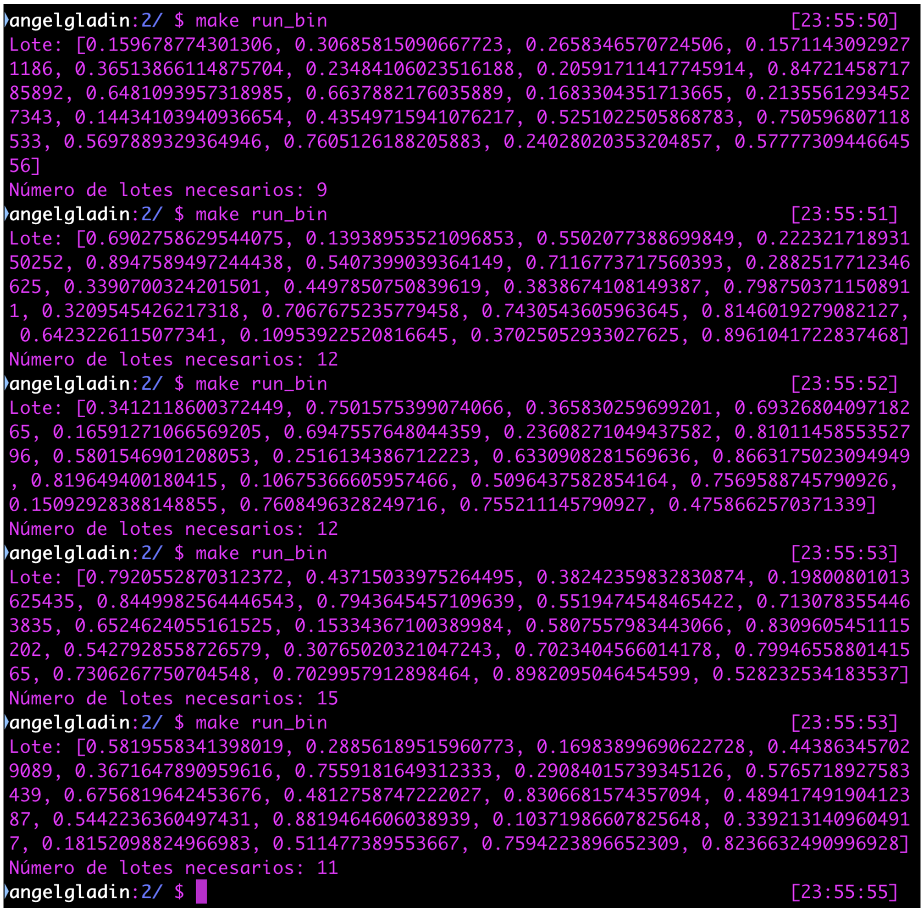
```
>>> bin_packing(items=[.85,.5,.4,.4,.3,.2,.2,.1], n=8, capacity=1)
4
>>> 
```

Figure 9.12 A result of applying nonincreasing first fit.

Sizes: 0.85, 0.5, 0.4, 0.4, 0.3, 0.2, 0.2, 0.1



Ejecución arbitraria mostrando su aproximación



```
angelgladin:2/ $ make run_bin                                          [23:55:50]
Lote: [0.159678774301306, 0.30685815090667723, 0.2658346570724506, 0.1571143092927
1186, 0.36513866114875704, 0.23484106023516188, 0.20591711417745914, 0.84721458717
85892, 0.6481093957318985, 0.6637882176035889, 0.1683304351713665, 0.2135561293452
7343, 0.14434103940936654, 0.43549715941076217, 0.5251022505868783, 0.750596807118
533, 0.5697889329364946, 0.7605126188205883, 0.24028020353204857, 0.57777309446645
56]
Número de lotes necesarios: 9
angelgladin:2/ $ make run_bin                                          [23:55:51]
Lote: [0.6902758629544075, 0.13938953521096853, 0.5502077388699849, 0.222321718931
50252, 0.8947589497244438, 0.5407399039364149, 0.7116773717560393, 0.2882517712346
625, 0.3390700324201501, 0.4497850750839619, 0.3838674108149387, 0.798750371150891
1, 0.3209545426217318, 0.7067675235779458, 0.7430543605963645, 0.8146019279082127,
 0.6423226115077341, 0.10953922520816645, 0.37025052933027625, 0.8961041722837468]
Número de lotes necesarios: 12
angelgladin:2/ $ make run_bin                                          [23:55:52]
Lote: [0.3412118600372449, 0.7501575399074066, 0.365830259699201, 0.69326804097182
65, 0.16591271066569205, 0.6947557648044359, 0.23608271049437582, 0.81011458553527
96, 0.5801546901208053, 0.2516134386712223, 0.6330908281569636, 0.8663175023094949
, 0.819649400180415, 0.10675366605957466, 0.5096437582854164, 0.7569588745790926,
0.15092928388148855, 0.7608496328249716, 0.755211145790927, 0.4758662570371339]
Número de lotes necesarios: 12
angelgladin:2/ $ make run_bin                                          [23:55:53]
Lote: [0.7920552870312372, 0.43715033975264495, 0.38242359832830874, 0.19800801013
625435, 0.8449982564446543, 0.7943645457109639, 0.5519474548465422, 0.713078355446
3835, 0.6524624055161525, 0.15334367100389984, 0.5807557983443066, 0.8309605451115
202, 0.5427928558726579, 0.30765020321047243, 0.7023404566014178, 0.79946558801415
65, 0.7306267750704548, 0.7029957912898464, 0.8982095046454599, 0.528232534183537]
Número de lotes necesarios: 15
angelgladin:2/ $ make run_bin                                          [23:55:53]
Lote: [0.5819558341398019, 0.28856189515960773, 0.16983899690622728, 0.44386345702
9089, 0.3671647890959616, 0.7559181649312333, 0.29084015739345126, 0.5765718927583
439, 0.6756819642453676, 0.4812758747222027, 0.8306681574357094, 0.489417491904123
87, 0.5442236360497431, 0.8819464606038939, 0.10371986607825648, 0.339213140960491
7, 0.18152098824966983, 0.511477389553667, 0.7594223896652309, 0.8236632490996928]
Número de lotes necesarios: 11
angelgladin:2/ $                                                       [23:55:55]
```

Integrante(s)

- Ángel Iván Gladín García - *(angelgladin@ciencias.unam.mx)*