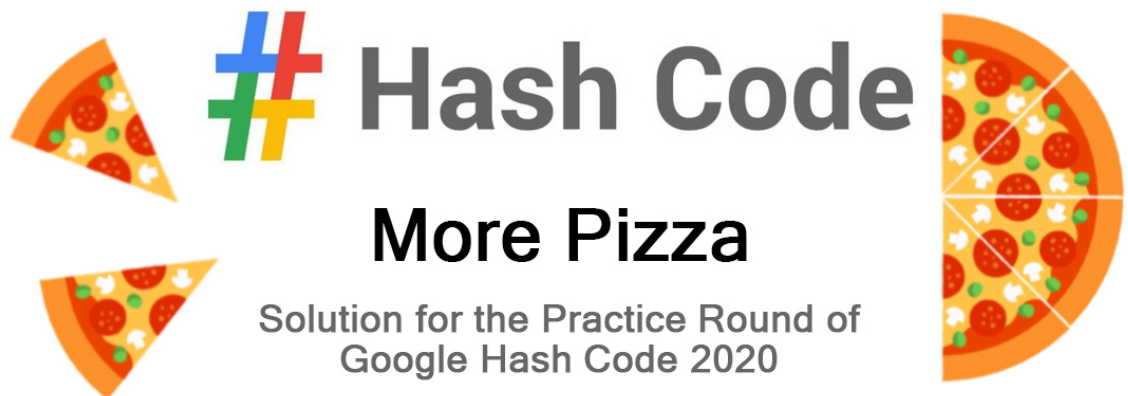


Proyecto 2 – Mas Pizza

DESCRIPCIÓN DEL PROBLEMA



Mas Pizza (o More Pizza) es un problema utilizado en una competición de Hash Code organizado por Google. Esta organizado en un centro Hash Code y se quiere pedir una pizza para los participantes. Afortunadamente hay una pizzería cercana. Existen diferentes tipos de pizza pero solo se puede pedir como máximo una pizza de cada tipo. Cada pizza tiene un tamaño específico siendo el tamaño el número de porciones de una pizza. Para calcular el máximo número de porciones que podemos pedir tendremos en cuenta el número de participantes registrados. Para reducir el desperdicio de alimentos, el objetivo es pedir la mayor cantidad de porciones de pizza como sea posible pero no más que el número máximo. El objetivo solicitar pizzas de diferentes siendo el número total de porciones menor o igual al número de participantes. El problema tiene 2 datos de entrada:

- **Porciones de pizza:** La cantidad de porciones de pizza necesarias.
- **Pizzas:** Un vector con las porciones para cada pizza.

Ejemplo:

- **Porciones necesarias:** 17
- **Pizzas** = c (2,5,6,8)

Una posible solución: Comprar 1 pizza de tipo 1 (2 porciones), 1 pizza tipo 3 (3 porciones) y pizza tipo 4 (8 porciones). El total sería 16 porciones y nos quedamos una por debajo del valor máximo.

El algoritmo nos debería devolver (elegir entre las dos opciones):

- Los índices de las pizzas que debo comprar
- El tamaño en secuencia

Nuestra tarea consistirá en implementar los algoritmos:

- Random Restart Hill Climbling
- Local beam search

Una vez implementados analizaremos los resultados

ANALISIS DE LOS RESULTADOS

Random restart Hill climbing

Small

	name	cost	runtime	iterations	max_depth
1	Random Restart Hill Climbing - It: 101	2.26	secs	8	3
2	Random Restart Hill Climbing - It: 100	0.02	secs	5	2
3	Random Restart Hill Climbing - It: 100	0.01	secs	1	3
4	Random Restart Hill Climbing - It: 100	0.01	secs	1	4
5	Random Restart Hill Climbing - It: 99	1.12	secs	8	4
6	Random Restart Hill Climbing - It: 99	1.11	secs	8	3
7	Random Restart Hill Climbing - It: 100	0.01	secs	2	4
8	Random Restart Hill Climbing - It: 99	2.19	secs	8	4
9	Random Restart Hill Climbing - It: 100	0.02	secs	5	3
10	Random Restart Hill Climbing - It: 100	0.00	secs	1	3

Medium

	name	cost	runtime	iterations	max_depth
1	Random Restart Hill Climbing - It: 4500	0.35	secs	7	23
2	Random Restart Hill Climbing - It: 4501	0.44	secs	8	22
3	Random Restart Hill Climbing - It: 4501	0.41	secs	8	20
4	Random Restart Hill Climbing - It: 4501	0.42	secs	8	19
5	Random Restart Hill Climbing - It: 4500	0.22	secs	4	21
6	Random Restart Hill Climbing - It: 4500	0.19	secs	4	15
7	Random Restart Hill Climbing - It: 4500	0.35	secs	7	16
8	Random Restart Hill Climbing - It: 4500	0.15	secs	3	20
9	Random Restart Hill Climbing - It: 4500	0.14	secs	3	19
10	Random Restart Hill Climbing - It: 4501	0.42	secs	8	18

Local beam search

K= 3

small

		name	cost	runtime	iterations	max_depth
1	Local Beam Search	115	0.06	secs	100	2
2	Local Beam Search	109	0.05	secs	100	2
3	Local Beam Search	125	0.06	secs	100	2
4	Local Beam Search	108	0.05	secs	100	2
5	Local Beam Search	105	0.06	secs	100	2
6	Local Beam Search	97	0.05	secs	100	2
7	Local Beam Search	126	0.06	secs	100	2
8	Local Beam Search	117	0.05	secs	100	2
9	Local Beam Search	125	0.06	secs	100	2
10	Local Beam Search	99	0.05	secs	100	2

Medium

		name	cost	runtime	iterations	max_depth
1	Local Beam Search	2690	0.05	secs	100	2
2	Local Beam Search	2908	0.07	secs	100	2
3	Local Beam Search	3173	0.06	secs	100	2
4	Local Beam Search	2606	0.06	secs	100	2
5	Local Beam Search	2757	0.06	secs	100	2
6	Local Beam Search	3380	0.06	secs	100	2
7	Local Beam Search	3189	0.06	secs	100	2
8	Local Beam Search	2767	0.06	secs	100	2
9	Local Beam Search	3241	0.07	secs	100	2
10	Local Beam Search	3070	0.06	secs	100	2

K= 5

Small

		name	cost	runtime	iterations	max_depth
1	Local	Beam Search	97	0.06 secs	100	2
2	Local	Beam Search	84	0.06 secs	100	2
3	Local	Beam Search	101	0.05 secs	100	2
4	Local	Beam Search	100	0.01 secs	2	2
5	Local	Beam Search	110	0.06 secs	100	2
6	Local	Beam Search	84	0.05 secs	100	2
7	Local	Beam Search	101	0.06 secs	100	2
8	Local	Beam Search	98	0.05 secs	100	2
9	Local	Beam Search	100	0.00 secs	2	2
10	Local	Beam Search	98	0.05 secs	100	2

Medium

			name	cost	runtime	iterations	max_depth
1	Local	Beam	Search	3057	0.07 secs	100	2
2	Local	Beam	Search	3260	0.06 secs	100	2
3	Local	Beam	Search	2942	0.07 secs	100	2
4	Local	Beam	Search	3033	0.07 secs	100	2
5	Local	Beam	Search	3212	0.06 secs	100	2
6	Local	Beam	Search	2877	0.07 secs	100	2
7	Local	Beam	Search	3136	0.06 secs	100	2
8	Local	Beam	Search	2775	0.07 secs	100	2
9	Local	Beam	Search	2808	0.07 secs	100	2
10	Local	Beam	Search	3474	0.06 secs	100	2

K=10**Small**

			name	cost	runtime	iterations	max_depth
1	Local	Beam	Search	97	0.07 secs	100	2
2	Local	Beam	Search	101	0.06 secs	100	2
3	Local	Beam	Search	100	0.02 secs	2	2
4	Local	Beam	Search	100	0.01 secs	2	2
5	Local	Beam	Search	115	0.06 secs	100	2
6	Local	Beam	Search	101	0.07 secs	100	2
7	Local	Beam	Search	97	0.07 secs	100	2
8	Local	Beam	Search	104	0.06 secs	100	2
9	Local	Beam	Search	100	0.01 secs	2	2
10	Local	Beam	Search	98	0.07 secs	100	2

Medium

			name	cost	runtime	iterations	max_depth
1	Local	Beam	Search	3599	0.07 secs	100	2
2	Local	Beam	Search	2940	0.07 secs	100	2
3	Local	Beam	Search	2926	0.08 secs	100	2
4	Local	Beam	Search	3042	0.08 secs	100	2
5	Local	Beam	Search	3183	0.07 secs	100	2
6	Local	Beam	Search	3324	0.07 secs	100	2
7	Local	Beam	Search	2997	0.08 secs	100	2
8	Local	Beam	Search	3036	0.08 secs	100	2
9	Local	Beam	Search	3101	0.07 secs	100	2
10	Local	Beam	Search	2943	0.07 secs	100	2

COMPARACIÓN DE RESULTADOS

RANDOM RESTART HILL CLIMBING				
	Cost	Runtime	Iterations	Max-depth
Tipo : Small	-	-	-	-
Media	99,8	0,675	4,7	3,3
Desviación típica	0,6	0,885	3,03	0,64
Tipo : Medium	-	-	-	-
Media	4500,4	0,3	6	19,3
Desviación típica	0,489	0,114	2,09	2,36
LOCAL BEAM SEARCH				
K = 3	-	-	-	-
Tipo : Small	-	-	-	-
Media	112,6	0,055	100	2
Desviación típica	10,16	0	0	0
K = 3	-	-	-	-
Tipo : Medium	-	-	-	-
Media	2978,1	0,061	100	2
Desviación típica	266,9525343	0,005676462122	0	0
K = 5	-	-	-	-
Tipo : Small	-	-	-	-
Media	97,3	0,045	100	2
Desviación típica	7,874713399	0,02173067468	0	0
K = 5	-	-	-	-
Tipo : Medium	-	-	-	-
Media	3057,4	0,066	100	2
Desviación típica	219,3648407	0,005163977795	0	0
K = 10	-	-	-	-
Tipo : Small	-	-	-	-
Media	101,3	0,045	100	2
Desviación típica	5,25039681	0,02838231061	0	0
K = 10	-	-	-	-
Tipo : Medium	-	-	-	-
Media	3109,1	0,074	100	2
Desviación típica	211,8261604	0,005163977795	0	0

CONCLUSIONES

Como podemos observar en la tabla de arriba , el tiempo de ejecución de el algoritmo Local beam search frente al random restart hill climbing es menor, ya que es un algoritmo mas probable en encontrar una mejor solución.

El algoritmo beam search trabaja con k estados actuales y simula un avance en paralelo con k estados actuales.

Mientras que en el random restart hill climbing, cada proceso de búsqueda se ejecuta independientemente de los demás, en el local beam search se abandona rápidamente aquellas búsquedas que no llevan a ninguna parte y se enfoca en los caminos donde habrá un mayor progreso.