Backtracking aplicado a la resolución de un problema de Bingo

NICOLÁS E. OLIVARES GONZÁLEZ[1,*]

[1] *Ingeniería Civil en Informática, Departamento de Ingeniería Informática, Universidad de Santiago de Chile*
[*] *Correo electrónico del autor: nicolas.olivares.g@usach.cl*

Compiled October 20, 2016

En el marco de la asignatura de Algoritmos Avanzados, impartida por el Departamento de Ingeniería Informática de la Universidad de Santiago, este artículo presenta el análisis de la solución para un problema presentado por el enunciado del laboratorio número 2 de la asignatura. Se aborda el uso de las técnicas de resolución de problemas en base a algoritmos.

OCIS codes:

http://dx.doi.org/10.1364/optica.XX.XXXXXX

1. INTRODUCCIÓN

Este artículo aborda los conceptos estudiados en cursos anteriores de algoritmos, se basa en el enfoque de resolución de problemas con el uso de algoritmos de backtracking o como son mejor conocido por la comunidad en general en su traducción al español, algoritmos de *Vuelta Atrás*.
En este artículo el tema a tratar se enfoca en la resolución del mismo problema abordado en el primer laboratorio, en el cuál se aplicó la técnica de enumeración explícita para la resolución de un problema de Bingo. Para el desarrollo de este laboratorio se utiliza el lenguaje de programación C y para la compilación del programa se trabaja en el ambiente de Linux.

2. DESPRICIÓN DEL PROBLEMA

La marca de juegos de azar Bingo! tiene como característica en sus juegos la elección de números entre el 1 al 20. El jugador ganador solo debe acertar a 8 de estos números. Esta gran cantidad de combinaciones representa una dificultad a la hora de realizar los juegos, por lo que la marca busca automatizar la obtención de estas combinaciones en base al uso de técnicas computacionales.

3. MARCO TEÓRICO

Al momento de solucionar un problema, se plantean diversas maneras o metodos de resolución en base a las características del problema. Cómo se vio en el laboratorio previo, el uso de enumeración descriptiva sirve para resolver problemas de este tipo, donde intervienen conjuntos de números que deben ser combinados de tal manera que algunas de las combinaciones de números obtenidas son parte de la solución del problema.

Los algoritmos de backtracking actúan sobre problemas en los cuales se cuentan con restricciones especificas que, durante la formación de las posibles soluciones, permiten descartar aquellos caminos donde ya no es posible seguir construyendo soluciones nuevas. Los algoritmos de backtracking son conocidos como *Vuelta atrás* porque si dado un punto del algoritmo en el cuál ya no es posible encontrar más soluciones con lo que se tiene temporalmente en el algoritmo, este vuelve atrás de manera que siga buscando las opciones que quedan pendientes a revisar.
Combinaciones sin repetición o combinaciones ordinarias de m elementos tomados de n en n (de orden n) son los distintos grupos de n elementos distintos que se pueden hacer con los m elementos que tenemos, de forma que dos grupos se diferencian en algún elemento y no en el orden de colocación. Se representa por Cm,n. ($n \geq m$).[1]

4. DESCRIPCIÓN DE LA SOLUCIÓN

Teniendo en consideración que el único dato de entrada proporcionado es un número entero $n > 8$, se describe la idea de solución.

A. IDEA

- Partiendo desde el numero n
- Se utiliza un enfoque recursivo
- Se tiene un conjunto de n números, en un comienzo con numeros del 1 al n
- Se tiene un conjunto, tupla o lista que se utiliza para introducir los valores posibles a medida que hacen los llamados recursivos.

- Ahora bien, en base a la idea principal es que se genere el arbol decisiones agregando los numeros del conjunto, y la condición entregada por la restriccion del problema es que los niveles del arbol no deben superar los 8 números de la combinación, en el momento en que se alcanza el nivel 0 significa que se alcanzó una combinación. Dado que los elementos se van introduciendo los elementos en la tupla, esta contendrá la combinación en ese instante.

Primero que nada el algoritmo de resolcuión cuenta de dos partes, una entrada al algoritmo en el cual se inicializan elementos necesarios para ejecutar el llamdo recursivo y así como también la parte recursiva en la que se desarrolla el resultado. En el algoritmo 1 se encuentra el algoritmo de entrada correspondiente al inicio del proceso.

Algorithm 1. Algoritmo entrada

```
1: procedure ALGORITMO_ENTRADA(n)
2:     conjunto = 1:n
3:     tupla = iniciarTupla()
4:     tamaño_comb = 8
5:     indice_partida = 0
6:     back_bingo(n,tamaño_comb,indice_partida,tupla,conjunto)
```

Algorithm 2. Backtracking bingo

```
1: procedure BT_BINGO(n,tamano_comb,indice,tupla,conjunto)
2:     if tamao_comb == 0 then
3:         mostrar o imprimir tupla return
4:     for i ← indice to n do
5:         insertar_final(tupla, conjunto[i])
6:         BT_BINGO(n,tamano_comb-1,i+1, tupla, conjunto)
7:         sacar_final(tupla)
       return
```

5. ANÁLISIS DE LOS RESULTADOS

En esta sección se realizará un análisis del algoritmo implementado en base a la serie de preguntas habituales que se realizan para comprobar la validez de un algoritmo.

A. Análisis: Preguntas para el Algoritmo 2

1. ¿El algoritmo para en algún momento?
   Si, el algoritmo para cuando todas los llamados recursivos han sido resueltos y no existan más combinaciones válidas posibles.

2. Y si para, ¿lo hace con la solución?
   Este algoritmo entrega la solución óptima para el problema, teniendo que dado un conjunto de números, el algoritmo entregara todas las combinaciones posibles entre sus elementos sin repeticion y en orden.

3. ¿Es eficaz?¿Soluciona el problema?
   El algoritmo es eficaz porque entrega las combinaciones pedidas por la marca de Bingo

4. ¿Es eficiente?
   Su orden de complejidad queda dado por la entrada n de manera O(n^n). En base al teorema maéstro donde $T(n) = nT(n-1) + O(1)$, se realizan n llamados recursivos al algorimto

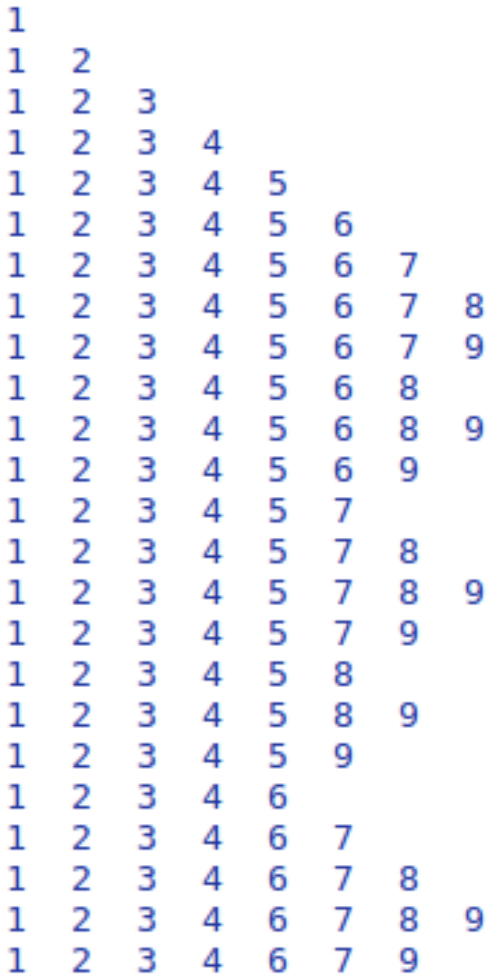
5. ¿Se puede mejorar?
   Se puede mejorar utilizando otra estrategia de descomposición, talvez con programación dinámica.

6. ¿Existe otro método para solucionar el problema?
   Existe la posibilidad de utilizar técnicas de programación dinámica o paralelización para realizar la tarea.

6. TRAZA DE LA SOLUCIÓN

La Figura 1 muestra una traza obtenida mediante el programa implementado, en donde es posible observar cada uno de los estados de la tupla que contiene a los elementos del conjunto combinados. Es decir, que en cada línea se presenta una tupla, es posible observar en las primeras lineas cómo se forma el árbol de decisión. En las líneas que tienen 8 elementos, se encuentra una combinación exitosa, el árbol se forma a partir del conjunto, pero no contiene más alla de 8 elementos en una de sus tuplas ya que la condición de corte o poda del algoritmo de backtracking da vuelta atrás y sigue con la operación pendiente que sería avanzar al siguiente elemento del conjunto o bien a retirar un elemento para subir un nivel en el árbol.

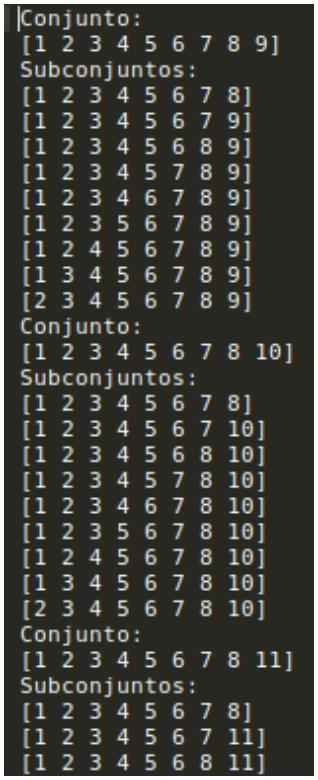
Fig. 1. Explicación de la traza para una entrada de $n = 9$



En el programa generado, se puede obtener una idea del funcionamiento del algoritmo.

Para el Algoritmo 2 se pueden revisar los resultados en el archivo *SALIDA.TXT*

Fig. 2. Demostración de algoritmo 2



La Figura 2 muestra el resultado del Algoritmo 2

7. CONCLUSIONES

Tras la realización del laboratorio y análisis de resultados, se concluye los siguiente:

- El algoritmo 2 logra su cometido, encontrando las combinaciones necesarias dado un conjunto de números.
- Si se itera el Algoritmo 2 puede acabar con las combinaciones restantes para los conjuntos desde n hasta 20
- El algoritmo hace una verificación o comprobación de criterio, también conocida como poda cuando el nivel alcanzado supera los 8 elementos, siendo ésta una característica o restricción del problema mismo.
- La aplicación de técnicas algoritmicas para la resolución de problemas implica el conocimiento de los diferentes tipos de recursos disponibles, incluso aquellos que no son eficientes y no son usados por la gran mayoría de los entendidos en el tema.
- Para la generación del programa se utilizó un TDA de TUPLA, el cuál consiste en una serie de algoritmos para manejar la isnerción y eliminación de elementos de esta. En sí es un manejo de listas enlazadas, pero con aglunas modificaciones para que las operaciones hechas sobre el TDA sean de O(1) para lograr que no afectaran en el desempeño del algoritmo principal que era el backtracking.

REFERENCES

1. L. B. Calmaestra, "Unidad didÁctica: Combinatoria. combinaciones sin repeticiÓn." (2016).