

Parte C: Backtracking, Branch and Bound

1. Se tienen K, a_1, \dots, a_n enteros no negativos.
 - a) Implementar un algoritmo usando la técnica Backtracking para determinar si existe un subconjunto de los $\{a_i\}$ tales que sumen exactamente K .
 - b) Implementar un algoritmo Branch and Bound para determinar el subconjunto de mayor suma tal que la misma sea $\leq K$.
2. Adaptar la implementación por backtracking/branch and bound del problema clásico de la mochila (es decir, el 0-1 Knapsack problem) para los siguientes casos:
 - a) Cada ítem i puede aparecer una cantidad n_i de veces (los números n_i son parámetros).
 - b) Cada ítem i puede aparecer una cantidad no acotada de veces.
 - c) Estudie la complejidad de su implementación en ambos casos, tanto para el problema de decisión como para el de optimización.
3. Implementar un algoritmo mediante la técnica de backtracking para poder determinar todos los cliques (subgrafo completo maximal) de un grafo.
4. Sea G un grafo dirigido conexo con costos positivos en las aristas. Implementar la técnica de Branch and bound para determinar, dados dos vértices u y v , el camino simple de MAYOR costo entre ambos.
5. Se tiene un tablero de ajedrez de $n \times n$ casilleros, se busca colocar n reinas de manera tal de que no se ataquen. Diseñar un algoritmo con la técnica de backtracking para resolver este problema.