Práctico 1: Formulación de problemas

Cruce de río con pesos

Tres personas A, B, C y un bote D se encuentran en la misma orilla de un río. El río se puede cruzar únicamente en bote y debe haber al menos una persona a bordo para poder navegarlo. El bote tiene una capacidad de 100 kg y bajo ningún motivo se puede exceder. Los pesos de las personas son: 100, 60 y 40 kg para A, B y C, respectivamente. El objetivo es que todas las personas lleguen al otro lado del río en el menor número de cruces.

1. Formular este problema como un problema de búsqueda.
2. Graficar el espacio de estados y hallar una solución óptima. ¿Cuántos estados son alcanzables?

Cambio con monedas

Este problema consiste en dar una cierta cantidad de cambio en monedas de cierta denominación, de forma tal que el número de monedas dadas sea mínimo. En la mayoría de los sistemas monetarios, este problema es sencillo. Por ejemplo, para dar 37 de cambio con monedas de 1, 5, 10 y 15, bastaría con elegir iterativamente la moneda de mayor denominación posible hasta alcanzar el total, es decir, $15 + 15 + 5 + 1 + 1$ (5 monedas). Sin embargo, en sistemas monetarios no convencionales, este algoritmo podría fallar. Por ejemplo, para dar 37 de cambio con monedas de 1, 7, 15 y 31, se tiene $31 + 1 + 1 + 1 + 1 + 1 + 1$ (7 monedas) contra $15 + 15 + 7$ (3 monedas).

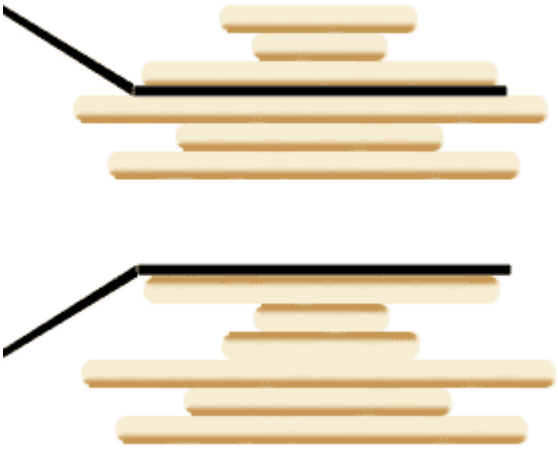
1. Formular este problema como un problema de búsqueda, considerando el caso particular de dar 6 de cambio y disponiendo de una cantidad ilimitada de monedas con denominación 1, 3 y 4.
2. Graficar el espacio de estados y hallar una solución óptima. ¿Cuántos estados son alcanzables?
3. Intentar formular este problema de forma general, es decir, para denominaciones arbitrarias. Para ello, serán de utilidad las siguientes notaciones: $X \in \mathbb{N}$ es el total de cambio a dar, $m \in \mathbb{N}$ es la cantidad de denominaciones y $d_i \in \mathbb{N}$ es el valor la i-ésima denominación para cada $i = 1, \ldots, m$. Por ejemplo, para el item 1, se tendría $X = 6$, $m = 3$, $d_1 = 1$, $d_2 = 3$ y $d_3 = 4$.

Torre de panqueques

Considerar una torre con n panqueques de diferentes tamaños, uno encima del otro sin ningún orden en particular. Se dispone de una espátula que puede ser insertada en

cualquier posición de la torre y permite voltear todos los panqueques agarrados con la misma.

Por ejemplo, los 3 panqueques de arriba de una torre se pueden voltear de la siguiente forma:

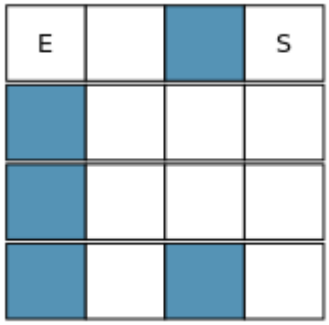


El objetivo de este problema es que los n panqueques queden ordenados en la torre de mayor a menor, en el menor número de pasos.

1. Formular este problema como un problema de búsqueda, intentando escribirlo de forma general para una torre de panqueques arbitraria. Para esto, serán de utilidad las siguientes notaciones: dada una torre de panqueques de entrada, para cada $i = 1, \ldots, n$, $d_i \in \mathbb{N}$ representa el diámetro en centímetros del i-ésimo panqueque, contando de abajo hacia arriba.
2. Sea una torre con $n = 3$, $d_1 = 10$, $d_2 = 12$ y $d_3 = 8$. Graficar el grafo de espacios de estados y hallar una solución óptima.

Escape del laberinto

Se tiene el siguiente laberinto en una grilla de tamaño 4×4, donde las casillas blancas representan espacios libres y las azules y el rectángulo exterior son paredes. El objetivo es encontrar un camino desde la casilla de entrada (E) del laberinto a la casilla de salida (S), de modo que la distancia total recorrida sea mínima.



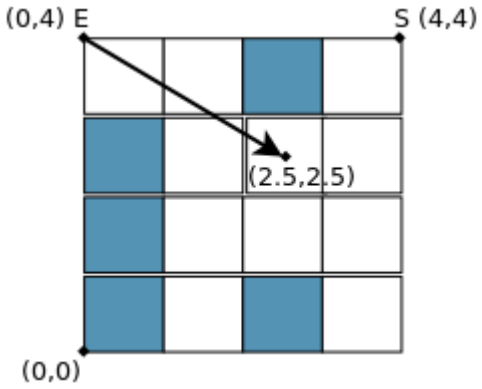
Considerar por el momento que únicamente es posible moverse de a una casilla a la vez en alguna de las 4 direcciones: arriba, abajo, derecha o izquierda, salvo que haya

una pared.

1. Formular este problema como un problema de búsqueda.

Ahora vamos a pensar que el laberinto se encuentra en un plano coordenado xy, donde el punto $(0, 0)$ se ubica en el extremo inferior izquierdo del laberinto, que cada casilla tiene tamaño 1×1 y que la entrada es el punto $(0, 4)$ y la salida el $(4, 4)$. En esta oportunidad, está permitido moverse en línea recta en cualquier dirección, sentido y módulo, siempre que no se atraviesen paredes.

Por ejemplo, podemos movernos en un paso desde la entrada al punto (2.5,2.5) de la siguiente forma:

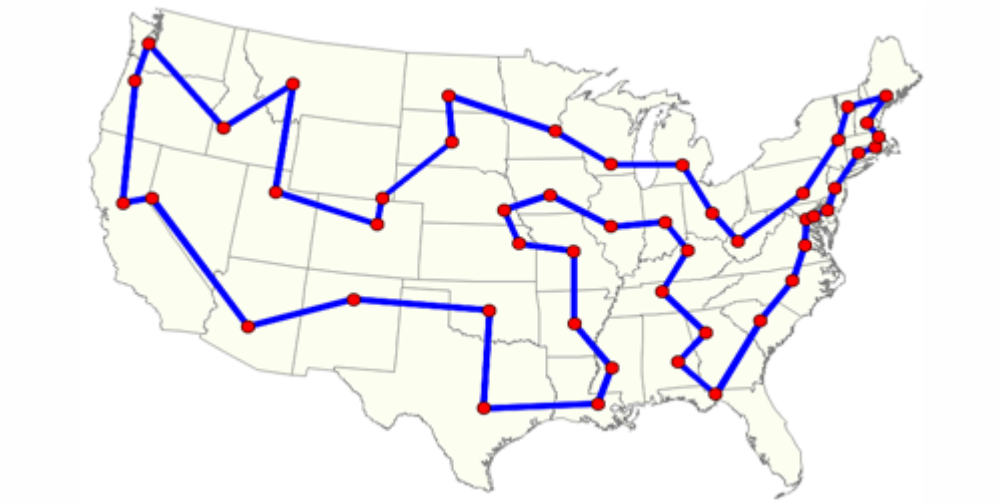


2. Proponer una formulación para esta nueva versión del problema. ¿Qué diferencias observa con la anterior?
3. Pensar en cuál sería ahora una solución óptima.
4. ¿Cómo redefiniría el espacio de estados considerando que toda solución óptima pasa por esquinas?

Viajante de comercio

Un viajante debe planificar un tour por n ciudades $\{v_1, \ldots, v_n\}$, partiendo de alguna de ellas y regresando a la misma, visitando exactamente una vez cada ciudad y minimizando la distancia total recorrida.

Por ejemplo, un tour óptimo por las ciudades capitales de EEUU es el siguiente.



Consideremos las siguientes alternativas para formular este problema.

- Cada estado del problema es una tupla con las ciudades visitadas hasta el momento. Por ejemplo, para $n = 5$, la tupla (v_1, v_4, v_3) representa que el viajante partió de v_1, luego viajó a v_4 y ahora se encuentra en v_3, debiendo aun visitar v_2 y v_5 y finalmente regresar a v_1. Sin pérdida de generalidad, podemos asumir que el viajante parte de v_1, así el estado inicial es (v_1). Las acciones consisten en viajar desde la última ciudad visitada a una ciudad no visitada, es decir, el resultado de la acción es agregar la nueva ciudad al final de la tupla. Por ejemplo, la acción viajar a v_2 en el estado (v_1, v_4, v_3) da como resultado el estado (v_1, v_4, v_3, v_2). Cuando falte visitar una única ciudad, digamos v_i, la acción agrega al final de la tupla a v_i y v_1 en un único paso. Por ejemplo, la acción viajar a v_5 en el estado (v_1, v_4, v_3, v_2) da como resultado el estado $(v_1, v_4, v_3, v_2, v_5, v_1)$.
- Cada estado del problema es una tupla con un tour parcial entre las ciudades visitadas hasta el momento. Por ejemplo, para $n = 5$, la tupla (v_1, v_4, v_3, v_1) representa que el viajante partió de v_1, luego viajó a v_4, luego viajó a v_3 y finalmente regresó a v_1. Sin pérdida de generalidad, podemos asumir que el viajante parte de v_1, así el estado inicial es (v_1, v_1). Las acciones consisten en agregar una ciudad no visitada entre cualquier par de ciudades del tour parcial. Por ejemplo, la acción de agregar la ciudad v_2 entre las ciudades v_1 y v_4 en el estado (v_1, v_4, v_3, v_1) da como resultado el estado $(v_1, v_2, v_4, v_3, v_1)$.

1. Dar dos formulaciones para el problema del viajante siguiendo las propuestas mencionadas. Suponer que el costo de viajar de una ciudad v_i a una ciudad v_j es igual a la distancia de viaje entre ellas y está dado por una función $\mathrm{dist}(v_i, v_j)$.
2. Considerando $n = 4$, ¿cuántos nodos tiene el árbol de búsqueda (considerando estados repetidos) en cada caso?
3. Para un valor de n grande, ¿cuál algoritmo de búsqueda general, en árboles o en grafos, usaría para resolver cada caso?

Corrector ortográfico

A partir de una palabra que no se encuentra en el diccionario, un corrector ortográfico se ocupa de sugerir palabras cercanas según ciertas métricas.

Dado un diccionario de palabras U y una palabra $s \notin U$, el objetivo de este problema es encontrar la palabra $u \in U$ tal que el número de acciones necesarias para convertir a s en u sea mínimo. Las posibles acciones son:

- Insertar una letra en alguna posición de la palabra,
- Eliminar una letra de la palabra e
- Intercambiar dos letras consecutivas de lugar.

Por ejemplo, si la palabra ingresada es acst y el universo es $U = \{\mathrm{casa}, \mathrm{gato}, \mathrm{perro}\}$, entonces la palabra más cercana sería casa y se obtiene, por ejemplo, primero eliminando la t (acs), luego insertando una a al final (acsa) y finalmente intercambiando a y c (casa).

1. Formular este problema como un problema de búsqueda.
2. ¿Hay caminos cíclicos en el espacio de estados? ¿Y redundantes?
3. ¿Cuántos nodos tiene el nivel 1 del árbol de búsqueda si el estado inicial es acst? ¿Y el nivel 2?