Evaluación asincrónica 3 de Estrategias Algorítmicas, ITESO, Luis Gatica

Fecha de entrega: sábado 14 de mayo de 2022

Nombre: Cordero Hernández, Marco Ricardo; Rodríguez Castro, Carlos Eduardo

Problemas de Programación Dinámica

Como respuesta a los siguientes problemas, incluya el código solución, una captura de pantalla de una ejecución exitosa (incluyendo entrada y salida), y un análisis en uno o dos párrafos de la complejidad algorítmica temporal y espacial de su solución. Para el análisis, pregunte cuáles parámetros deben ser considerados: los resultados no serán dependientes de una sola variable (como, en contraste, usualmente pensamos dados la notación *T(n)* y el hecho de que el tiempo de ejecución suele ser una función solamente del tamaño de la entrada).

1. Escriba un método estático que resuelva el problema del cambio a partir de una cantidad a entregar (ejemplo: 9) y de un arreglo con las denominaciones disponibles (ejemplo: monedas de 1, 4 y 6). La respuesta debe indicar cuántas monedas de cada denominación se usaron para entregar la cantidad final.

```java
public class Cambio {
    public static int min(int a, int b) {
        return (a > b || a == b) ? b : a;
    }

    public static int solve(int[] monedas, int cantidad) {
        int[] sols = new int[cantidad + 1];
        sols[0] = 0;
        for (int i = 1; i < cantidad + 1; i++) sols[i] = Integer.MAX_VALUE;

        for (int i = 1; i < cantidad + 1; i++) {
            for (int moneda : monedas) {
                if (i - moneda >= 0) {
                    sols[i] = min(sols[i], 1 + sols[i - moneda]);
                }
            }
        }

        return (sols[cantidad] != cantidad + 1) ? sols[cantidad] : -1;
    }

    public static void main(String[] args) {
        int cantidad = 9;
        int[] monedas = {1, 4, 6};

        System.out.println(solve(monedas, cantidad));
    }
}
```

2. Escriba un método estático que resuelva el problema de la mochila NO fraccionario a partir de un peso total que puede soportar la mochila y de un arreglo con los objetos disponibles (donde cada objeto está compuesto por un nombre, un peso y un valor). La respuesta debe indicar cuántas instancias de cada objeto fueron seleccionadas. En este caso, indique también cuál fue el valor total que acumuló la mochila.

```
class Mochila {
    static class Item {
        final String name;
        final int w, v;

        Item (String name, int w, int v) {
            this.name = name;
            this.w = w;
            this.v = v;
        }

        @Override
        public String toString() {
            return this.name + ": <Valor: " + this.v + ", Peso: " + this.w + ">";
        }
    }

    static int max(int A, int B) {
        return (A > B) ? A : B;
    }

    static void backpack(Item[] items, int W) {
        int n = items.length;
        int[] w = new int[items.length],
              v = new int[items.length];
        int[][] vals = new int[n + 1][W + 1];

        for (int a = 0; a < n; a++) {
            w[a] = items[a].w;
            v[a] = items[a].v;
        }

        for (int i = 0; i <= n; i++) {
            for (int j = 0; j <= W; j++) {
                vals[i][j] = (i == 0 || j == 0) ? 0 :
                             (w[i - 1] <= j) ? max(v[i - 1] + vals[i - 1][j - w[i - 1]], vals[i -
1][j]) :
                             vals[i - 1][j];
            }
        }

        int res = vals[n][W];
        System.out.println("Valor final: " + res);

        int holder = W;
        for (int i = n; i > 0 && res > 0; i--) {
            if (res == vals[i - 1][holder]) continue;
            else {
                System.out.println(items[i].toString());

                res -= v[i - 1];
                holder -= w[i - 1];
            }
        }
    }

    public static void main(String[] args) {
        Item i1 = new Item("Sacapuntas", 1, 1);
        Item i2 = new Item("Corrector", 2, 6);
        Item i3 = new Item("Calculadora", 5, 18);
        Item i4 = new Item("Teclado 20%", 6, 22);
```

```
        Item i5 = new Item("PC Gamer", 7, 28);

        Item[] items = {i1, i2, i3, i4, i5};
        int W = 11;

        backpack(items, W);
    }
}
```

Conclusiones generales del semestre

Rodríguez Castro: Con este trabajo y los demás de la materia pude desarrollar un pensamiento para partir el problema en pequeños problemas más fáciles de resolver. Este pensamiento me ayudará en las entrevistas, a resolver y calcular tiempos de ejecución y de espacio, algo que es muy útil en el ámbito de los algoritmos.

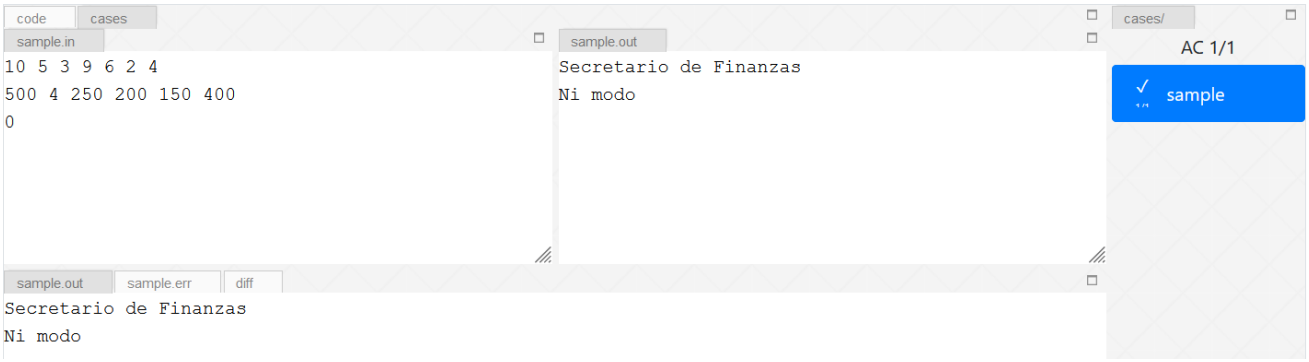
Cordero Hernández: El semestre resultó extenuante, arrebatando una gran porción de la pasión por la programación con la que entré a la carrera. Fue muy interesante revisar muchos algoritmos que a menudo se mencionan en el ámbito informático pero que en múltiples ocasiones no se conoce su funcionamiento ni utilidad. Me parece que pude haberle sacado más provecho al curso sin lugar a duda, sin embargo, dada el carácter de dificultad con el que cuenta el mismo no se podría rescatar algo más allá de lo analizado. Hubiera sido de sumo interés analizar más problemas reales con soluciones fundamentadas en los algoritmos revisados pero sin el impedimento que la presión de la calificación conlleva.

Problemas de concurso

- Gastos perplejos

```
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.util.Collections;
import java.util.Vector;

public class Main {
  static boolean caseSol;

  static void unique_combination(int l, int sum, int K, Vector<Integer> local, Vector<Integer> A) {
      if (sum == K) {
          caseSol = true;
          return;
      }

      for (int i = l; i < A.size(); i++) {
          if (sum + A.get(i) > K)
              continue;

          if (i > l && A.get(i) == A.get(i - 1) )
              continue;

          local.add(A.get(i));
          unique_combination(i + 1, sum + A.get(i), K, local, A);
          local.remove(local.size() - 1);
      }
  }

  static void Combination(Vector<Integer> input) {
      Vector<Integer> A = new Vector<>();
```

```
    int K = input.get(0);

    for (int i = 2; i < input.size(); i++) A.add(input.get(i));

    Collections.sort(A);
    Vector<Integer> local = new Vector<Integer>();
    unique_combination(0, 0, K, local, A);
  }

  static Vector<Integer> format(String s) {
    String[] f = s.split(" ");
    Vector<Integer> r = new Vector<>();

    for (int i = 0; i < f.length; i++) {
      r.add(Integer.parseInt(f[i]));
    }

    return r;
  }

  public static void main(String[] args) throws IOException {
    BufferedReader br = new BufferedReader(
                        new InputStreamReader(System.in));

    String caso = br.readLine();

    while(!caso.equals("0")) {
      caseSol = false;

      Combination(format(caso));
      System.out.println((caseSol) ? "Secretario de Finanzas" : "Ni modo");

      caso = br.readLine();
    }
  }
}
```



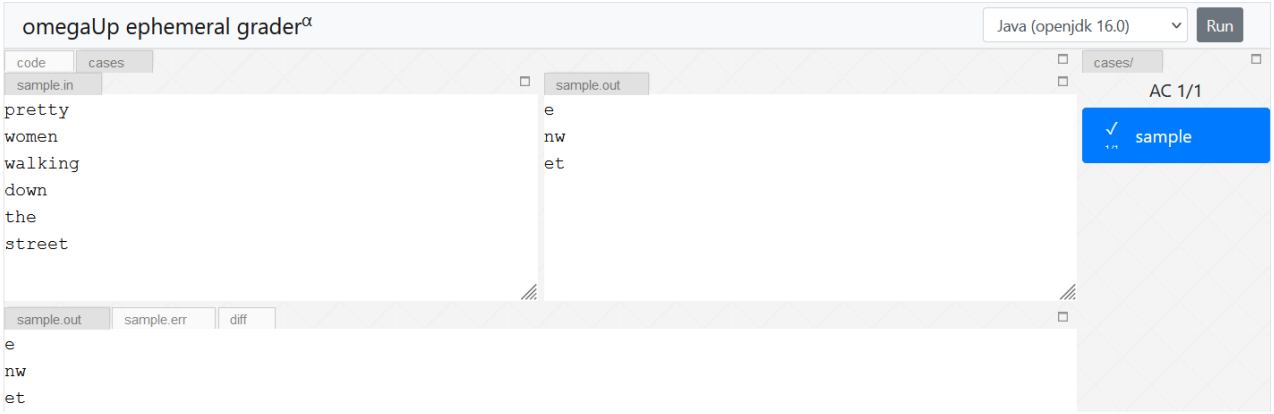
Date and Time	GUID	Status	Percentage	Language	Memory	Runtime	Actions
2022-05-14 13:19	e806cf57	TLE	47.62%	java	0.00 MB	>8.99 s	

- Permutación común

```
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.util.Arrays;

public class Main {
  static int max(int A, int B) {
```

```
    return (A > B) ? A : B;
}

static char[] solve(String s1, String s2) {
    int[][] dp = new int[s1.length() + 1][s2.length() + 1];

    for (int i = 0; i <= s1.length(); i++) {
        for (int j = 0; j <= s2.length(); j++) {
            if (i == 0 || j == 0)
                dp[i][j] = 0;
            else if (s1.charAt(i - 1) == s2.charAt(j - 1))
                dp[i][j] = dp[i - 1][j - 1] + 1;
            else
                dp[i][j] = max(dp[i - 1][j],
                               dp[i][j - 1]);
        }
    }

    int index = dp[s1.length()][s2.length()],
        i = s1.length(), j = s2.length();
    char[] sol = new char[index + 1];

    while (i > 0 && j > 0) {
        if (s1.charAt(i - 1) == s2.charAt(j - 1)) {
            sol[index - 1] = s1.charAt(i - 1);
            i--; j--; index--;
        } else if (dp[i - 1][j] > dp[i][j - 1]) {
            i--;
        } else j--;
    }

    return sol;
}

static void printSol(char[] sol) {
    String holder = "";

    for (char c : sol) holder += c;

    System.out.printf("%s\n", holder.substring(1));
}

public static void main(String[] args) throws IOException {
  BufferedReader br = new BufferedReader(
                        new InputStreamReader(System.in));

  String s1 = br.readLine(), s2 = br.readLine();

  while (s1 != null) {
    char[] sol = solve(s1, s2);
    Arrays.sort(sol);
    printSol(sol);

    s1 = br.readLine(); s2 = br.readLine();
  }
}
```

```
}
```



Date and Time	GUID	Status	Percentage	Language	Memory	Runtime	Actions
2022-05-14 13:20	7ce88e3a	TLE	0.00%	java	32.89 MB	>2.56 s	