

Ricardo Pineda
20160164

Laboratorio Algoritmia 6

Ejercicio 1

Fracciones Egipcias:

Cada fraccion positiva se puede representar como una suma de fracciones unitarias. Fracciones unitarias son aquellas donde el numerador es 1 y el denominador es un entero positivo.

Fraccion egipcia de $2/3 = 1/2 + 1/6$ Fraccion egipcia de $6/14 = 1/3 + 1/11 + 1/231$ Fraccion egipcia de $12/13 = 1/2 + 1/3 + 1/12 + 1/156$ Para una fraccion de forma nume/deno donde $\text{deno} \neq 0$, nume obtengan la mayor fraccion unitaria posible, y recurran hasta encontrarla.

Desarrollen un algoritmo codicioso que resuelva las fracciones egipcias y describan por que es codicioso.

Asumiendo que el valor que buscamos esta dado por T:

Algorithm 1 Fracciones Egipcias (Codicioso)

```
1: while fin != true do
2:   Aprox = Floor(Numerador/Denominador);
3:   Print "1/ " + Aprox + 1;
4:   Resultado = (Numerador/Denominador) - (1/(Aprox + 1));
5:   if Numerador(Resultado) != 1 then
6:     Numerador = Denominador(Resultado);
7:     Denominador = Numerador(Resultado);
8:   else
9:     print "Se encontro la mas cercana";
10:    fin = true;
```

El algoritmo es codicioso ya que el ciclo va generando fracciones unitarias. Tambien cabe mencionar que el algoritmo solo se preocupa por el resultado inmediato y no verifica que hayan otras combinaciones entre fracciones.

Ejercicio 2

Algoritmos Codiciosos + Dinamicos

Desarrollen un programa dinamico y otro codicioso para el problema de knapsack fraccionario. Encontrando el valor maximo.