

Tarea1 Análisis de Algoritmos

Montaño Pérez Joshua Said 317222812

29 AGOSTO 2022

1 Tener un problema bien definido

Determinar si un número w , menor que B es o no primo

- nombre del problema:
Ver si un número w , menor que B es o no primo
- Limites del problema:
Ver si el número w es primo
- Resultados deseados:
Qué w sea menor que B y sea primo

2 Analizar el problema

- Datos de entrada:
Necesitamos un número w natural
- Proceso:
Utilizaremos recursión:
Primera condición:
cuando w es menor a B y es primo regresamos que es TRUE
Segunda condición:
Queremos saber primero si w es menor que B , entonces tenemos que verificar que $(w-1) < B$, además de esa condición w debe de ser primo, en este caso utilizamos el mod de 2 que es 0 para verificar que w lo sea y por ultimo regresamos que True si las dos condiciones anteriores se cumplen.
- Salida:
Sabremos que w es menor que B y además es un número primo

3 Diseño del algoritmo

- Identificadores parametros y constantes
 w es un parametro
 B es una constante

- Requerimos para diseñar el algoritmo
Nuestra variable w y B
- Precondiciones:
Ya que w tiene q ser menor que B , $w < B$
- Postcondición:
Sabremos si w es menor y si es o no es primo
- Descripción del algoritmo:

1. Recibiremos una w
2. Ver si w es menor que B , en caso de que sea menor proseguimos a verificar si es primo (Este es nuestro Caso Base)
3. Ahora verificamos de que $w < B$ si es así seguimos al siguiente paso.
4. En el caso de que no lo sea regresamos FALSE, ya que no cumple la condición de w sea menor a B .
5. Después verificamos de que w sea primo, en este paso utilizamos el mod de 2 en w y el residuo nos debe de dar 0 y cuando esta condición se cumpla regresamos TRUE ya que las dos condiciones previas se cumplieron.
6. En el caso de que el residuo no sea 0, significa que w no es un número primo y por lo tanto regresaremos FALSE ya que no se cumple la condición de que w sea número primo

- Algoritmo en código:

```

1 public class ParedAlgoritmos {
2
3
4     public static boolean EsPrimoYMenor(int w){
5
6         int b = 45;
7         if(w < b){
8             for(int i = 2; i*i<= w; i++){
9                 if(0 == w % i){
10                     return false;
11                 }
12                 return true;
13             }
14         }else{
15             return false;
16         }
17         return true;
18     }
19 }

```

4 Verificación del algoritmo

1. Tomemos los valores de $w = 4$ y $B = 45$

2. Ahora verificamos que $4 < 45$ y esto se cumple proseguimos al siguiente paso.
3. Ahora pasamos a verificar al $\text{if}(0 == 4 \bmod 2)$, como 4 no es primo regresa FALSE

```
joshua@joshua-VivoBook-ASUS-Laptop-X505ZA-A505ZA:~/Documentos/AnalisisAlgoritmos$ javac Tarea1Algoritmos.java
joshua@joshua-VivoBook-ASUS-Laptop-X505ZA-A505ZA:~/Documentos/AnalisisAlgoritmos$ java Tarea1Algoritmos
false
joshua@joshua-VivoBook-ASUS-Laptop-X505ZA-A505ZA:~/Documentos/AnalisisAlgoritmos$
```

Con esta ejecución verificamos que el algoritmo es correcto y vemos que $4 < 45$, además de que no es primo.