Diseño de Algoritmos

Sesión 04

Profesores:

Tomás Lara Valdovinos – t.lara@uandresbello.edu

Jessica Meza-Jaque – je.meza@uandresbello.edu

OBJETIVOS DE LA SESIÓN

- Conocer concepto de eficiencia.
- Resumir brevemente lo aprendido a la fecha.
- Evaluar lo aprendido a la fecha:
  - concepto de algoritmo
  - problema-solución

CONTENIDOS DE LA SESIÓN

- Resumen con alumnos
- Breve ejercitación
- Aplicación Desafío 1

Eficiencia en algoritmos

- *“Al elegir un algoritmo para resolver un problema, estás intentando predecir qué algoritmo será más rápido para un grupo particular de datos en una plataforma específica (familia de plataformas). Caracterizar el tiempo esperado de computación de un algoritmo es inherente a un proceso matemático.”*

Heineman, G. et al (2009) *Algorithms in a Nutshell*. Ed. O’Reilly

Algoritmos y eficiencia

- Los algoritmos son importantes cuando hablamos de eficiencia.
- Saber elegir el algoritmo hace la diferencia en el software que estás produciendo.

- La recomendación es:
  - Utiliza estructuras de datos adecuadas
  - Entiende el problema, es decir,
    - Identifica posibles causas, y
    - Describe el problema

Algoritmos y eficiencia

- Si decides resolver el problema basado sólo en lo que tu crees que es la causa, podría ocurrir que:
  - Resuelvas el problema equivocado
  - No explores, posiblemente, mejores soluciones.

- Experimenta si es necesario:
  - Ejecuta el código bajo distintas circunstancias.
  - Explota las capacidades máximas del algoritmo.

Algoritmos y eficiencia

Ejemplo

Algoritmos y eficiencia - Ejemplo

ProgramA

```
int main(int argc, char **argv){
    int = 0;
    for(i = 0; i < 1000000; i++){
        malloc(32);
    }
    exit(0);
}
```

Algoritmos y eficiencia - Ejemplo

ProgramB

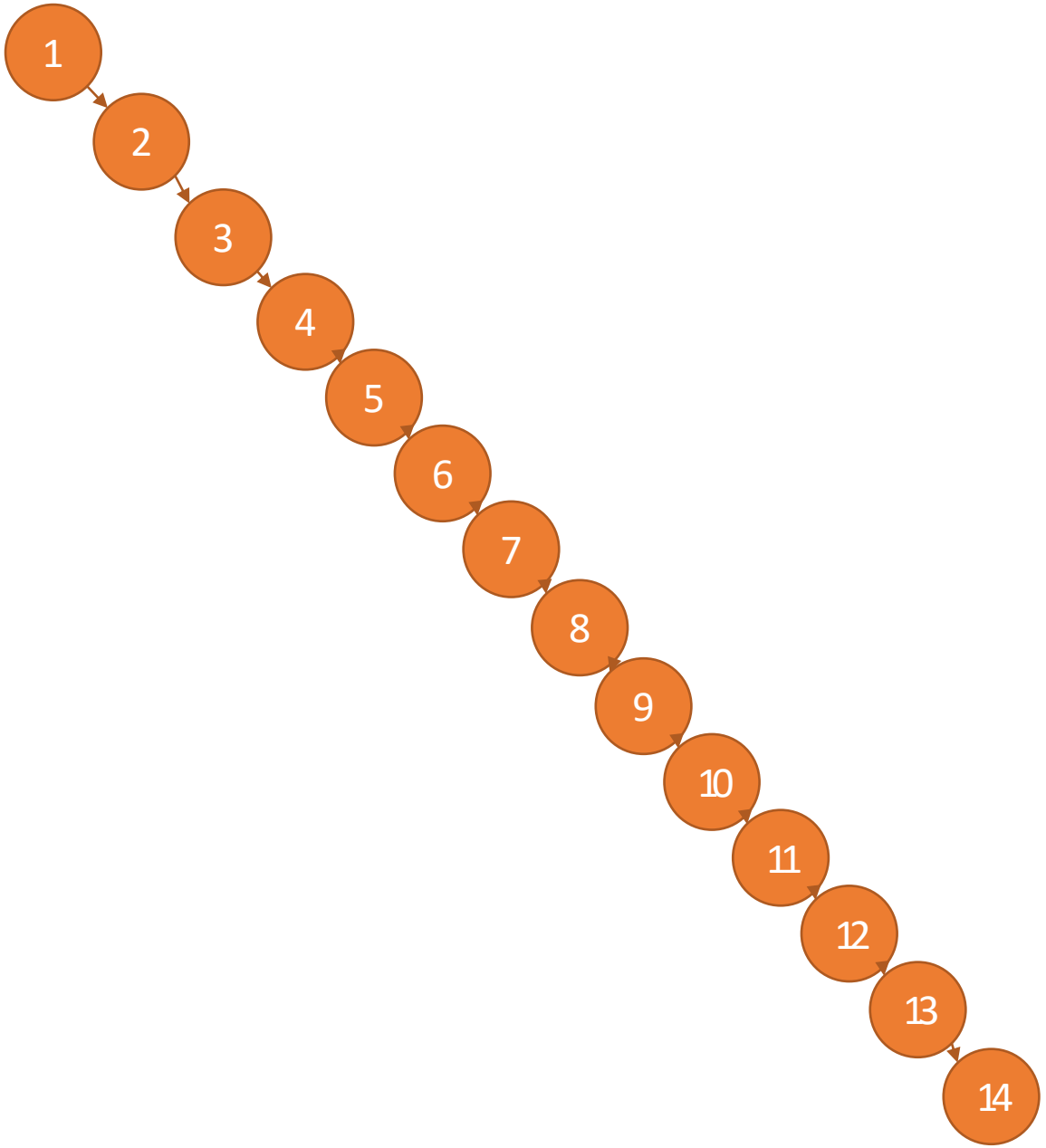
```
Int main(int argc, char **argv){
    int = 0;
    for(i = 0; i < 1000000; i++){
        void *x = malloc(32);
        free(x);
    }
    exit(0);
}
```

Algoritmos y eficiencia - Ejemplo

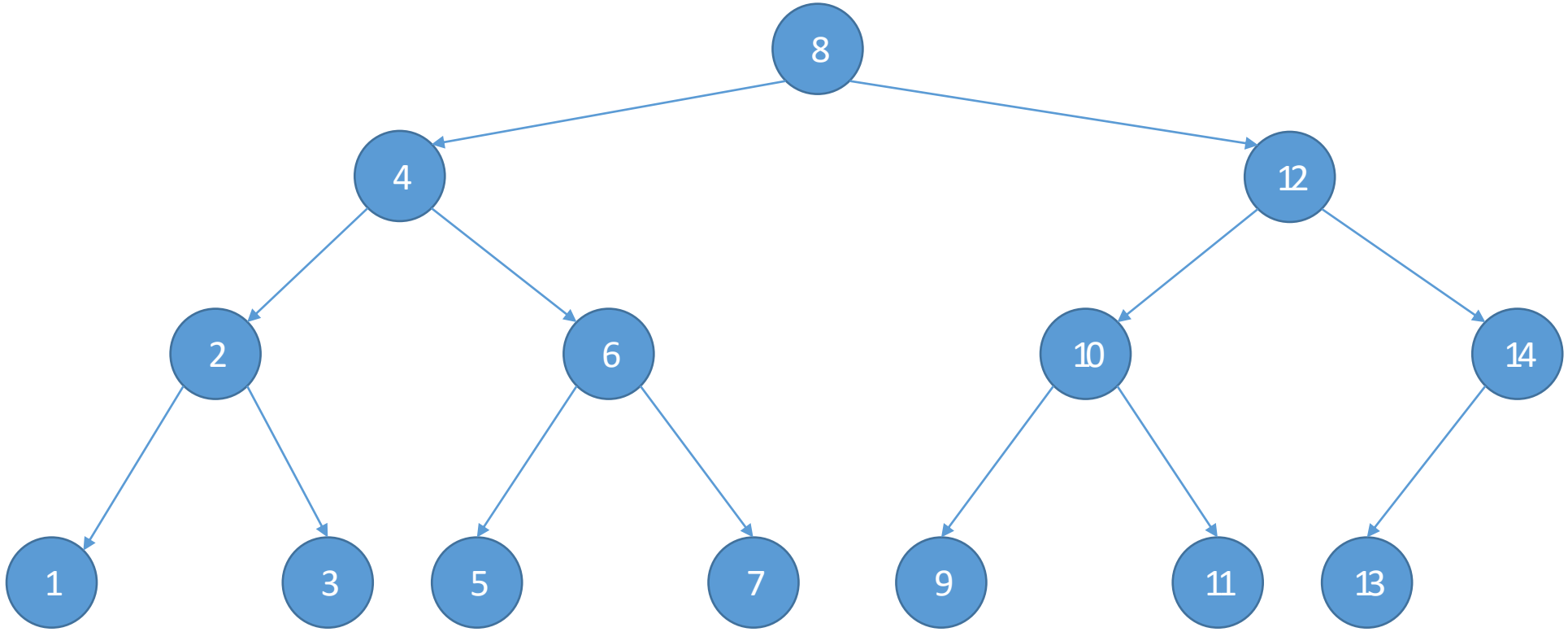
ProgramC

```
int main(int argc, char **argv){
   int = 0;
   void *addrs[1000000];
   for(i = 0; i < 1000000; i++){
      addrs[i] = malloc(32);
   }
   for(i = 0; i < 1000000; i++){
      free(addrs[i]);
   }
   exit(0);
}
```

Árbol sin balancear con nodos consecutivos



Mismo árbol binario, perfectamente balanceado



Árbol balanceado

- Permite asegurar búsqueda de nodos en tiempo logarítmico
- Se define profundidad de una hoja como la distancia entre un nodo hoja y la raíz.
- En un árbol perfectamente balanceado se cumple que:
  - Profundidad(H1) – Profundidad (H2) ≤ 1
  - Para cualquier par de nodos hoja H1 y H2
  - También, Profundidad(H_i) ≤ log(n), n = cantidad de nodos en el árbol

Árbol binario Rojo-negro

- Implementación eficiente de un árbol balanceado
- Se cumple que:
  - Profundidad(H1) / Profundidad(H2) ≤ 2
  - Para cualquier par de nodos hoja H1 y H2
  - También, Profundidad(H_i) ≤ 2 * log(n + 1)

Trabajo Grupal en Sala

Eficiencia: Tiempo y Recursos

FORMA EFICIENTE DE CELEBRAR CON PIZZAS?

Escenario: Reunión de amigos para ver el futbol (4 personas)

FORMA EFICIENTE DE CELEBRAR CON PIZZAS?

Escenario: Reunión de amigos para ver el futbol (4 personas)

Opciones a evaluar:

a) Hacer la/s pizza/s completa (masa y demás)
b) Comprar masa hecha y rellenar/hornear en casa
c) Comprar la/s pizza/s hechas

FORMA EFICIENTE DE CELEBRAR CON PIZZAS?

Escenario B: Reunión de amigos para ver el futbol (4 personas)

A más eficiente que B
B más eficiente que A

Opciones a evaluar:

a) Hacer la/s pizza/s completa (masa y demás)
b) Comprar masa hecha y rellenar/hornear en casa
c) Comprar la/s pizza/s hechas

TIEMPO

DINERO

FORMA EFICIENTE DE CELEBRAR CON PIZZAS?

OTRO ESCENARIO

Escenario: Cena para los suegros que gustan mucho de las pizzas (4 personas)

Resultados

- Las estructuras de datos son útiles para resolver problemas eficientemente
- Prueba posibles escenarios y mide su comportamiento
- Busca el algoritmo que cumpla las necesidades de tu aplicación de manera aceptable, no es necesario un algoritmo experto.
- No necesitas inventar técnicas de algoritmos nuevos, pero ten las existentes a mano para ver cual se ajusta mejor a tu solución.

Diseño de Algoritmos

Sesión 04

Profesores:

Tomás Lara Valdovinos – t.lara@uandresbello.edu

Jessica Meza-Jaque – je.meza@uandresbello.edu