

Algorísmia i Programació 1, GCED, 25 de gener de 2018

Possibles solucions

1 Pupurri

1. ...
2. Un compilador és un programa que tradueix un programa escrit en un llenguatge de programació a un programa equivalent en un altre llenguatge de programació tot verificant possibles errors.
3. Donat un natural n , el garbell d'Eratòstenes calcula tots els nombres primers entre 0 i n .
4. Els paràmetres formals són els que s'utilitzen per definir els cos de les funcions. Els paràmetres reals són els valors o variables que es donen al cridar les funcions.
5. Si el paràmetre és de sortida o d'entrada/sortida, pas per referència. Altrament, pas per valor si el tipus és petit, i pas per referència constant si és potencialment gran.
6. 256
7. 0
8. 255
9. C++: 4 7 7 4. Python3: 3.5 6.5 3 5
10. g++ -Wall -std=c++11 -O2 p.cc

2 Operacions geomètriques

```
bool dins (Punt p, Cercle c) {  
    return  $\text{sqr}(p.x - c.c.x) + \text{sqr}(p.y - c.c.y) \leq \text{sqr}(c.r)$ ;  
}  
  
Rectangle bounding_box (Cercle c) {  
    return {{c.c.x - c.r, c.c.y - c.r}, {c.c.x + c.r, c.c.y + c.r}};  
}  
  
void traslladar (Rectangle& r, double dx, double dy) {  
    r.ie.x += dx; r.ie.y += dy; r.sd.x += dx; r.sd.y += dy;  
}  
  
Punt aleatori (Rectangle r) {  
    return {rand_double(r.ie.x, r.sd.x), rand_double(r.ie.y, r.sd.y)};  
}  
  
Punt aleatori (Cercle c) {  
    Punt p = aleatori(bounding_box(c));  
    if (dins(p, c)) return p; else return aleatori(c);  
}
```

```

}

Rectangle bounding_box (Cercles& cercles) {
    Rectangle R;
    R.ie = R.sd = cercles[0].c;
    for (Cercle C : cercles) {
        R.ie.x = min(R.ie.x, C.c.x - C.r);
        R.ie.y = min(R.ie.y, C.c.y - C.r);
        R.sd.x = max(R.sd.x, C.c.x + C.r);
        R.sd.y = min(R.sd.y, C.c.y + C.r);
    }
    return R;
}

double rand_double (double a, double b) {
    return a + rand() / double(RAND_MAX) * (b - a);
}

```

Els *Punts*, *Rectangles* i *Cercles* es poden passar per referència constant o per valor.

3 Ordenació per bombolla

1. Els elements en posicions menors que i ja estan correctament ordenats.
2. $\sum_{i=0}^{n-2} \sum_{j=i+1}^{n-1} 1 = \frac{1}{2}(n^2 - n)$.
3. $O(n^2)$.

4 Ordenació de creps (*crêpes*, *pancakes*)

L'algorisme mantindrà les creps en dos blocs: el bloc inferior i el superior. En el bloc inferior, les creps ja estan correctament ordenades.

Començant amb el bloc inferior buit, cada cop es farà més gran, fins aconseguir que el bloc superior sigui buit i, per tant, totes les creps estiguin ordenades.

Per fer que el bloc inferior es faci una crep més gran, cal aplicar dos girs d'espàtula: El primer sota la crep més gran del bloc superior (de forma que aquesta queda al damunt), i el segon gir entre la frontera dels dos blocs.

Cal fer, doncs, $2(n - 1)$ girs com a molt.

5 Depuració de codi

(2 punts)

```

vector<Llibre> llegir () {
    Llibre llibre;      M: declarar dins del bucle
    int n;
    cin >> n;
    vector<Llibre> llibres(n);    E: deixar vector buit o no fer push-back despres
    for (int i = 0; i <= n; ++i)    E: i < n
        cin >> llibre.titol >> llibre.autor >> llibre.pagines;
}

```

```

        llibres.push_back(llibre);
    }
    return llibres;
}

```

ε: ha de retornar bool

```

void comp (Llibre l1, Llibre l2) {      M: passar per referencia constant
    ε: cal invertir les dues línies següents
    if (l1.titol ≠ l2.titol) return l1.titol ≤ l2.titol;
    if (l1.autor ≠ l2.autor) return l1.autor ≤ l2.autor;
    return (l1.pagines ≥ l2.pagines);
    M: no calen parentesis
    ε: hauria de ser >
}

```

M: per coherencia amb les demes: ordenar

```

void ordena (vector<Llibre> llibres) {      ε: passar per referencia
    sort(llibres.begin(), llibres.end(), comp);
}

```

```

void escriure (vector<Llibre> llibres) {      M: passar per referencia constant
    for (Llibre llibre : llibres) {
        cout << llibre.titol << " " << llibre.autor << " " << llibre.pagines << endl;
    }
}

```

```

void main()      ε: ha de retornar int
    vector<Llibre> llibres = llegir();
    ordena(llibres);
    escriure(llibres);
}

```