La velocitat de correcció del Jutge.org depèn del cost del algorisme?

Borja Arias Navarro, Javier Calvo Martín, Marc Gallofré Ocaña
Probabilitat i Estadística, grup 11
Maig 2012

Resum

Objectiu:

L'objectiu d'aquest experiment és veure si influeix el cost d'un algorisme en C++ en el temps que triga la web Jutge.org a donar el seu veredicte d'execució.

Hipòtesis:

Creiem que amb algorismes de costos majors tardarem més en obtenir un veredicte que amb algorismes de costos menors. En concret l'algorisme de la Cerca Dicotòmica s'executarà i obtindrà el veredicte en el jutge abans que l'algorisme de les Torres de Hanoi.

Mètode:

Hem implementat els dos algorismes (Cerca Dicotòmica i Torres de Hanoi) i primer de tot hem verificat que els seus costos siguin clarament ben diferents. Posteriorment s'han realitzat un total de 60 enviaments (30 per la Cerca Dicotòmica i 30 per Torres de Hanoi), tots dos algorismes correctes i obtenint un veredicte d'execució correcte (semàfor verd), amb els quals hem fer una recollida de dades (temps), mesurades des del moment en que es puja l'arxiu fins que apareix el semàfor. I per tal de posar en contrast la nostre hipòtesis, em enfrontat les nostres dades.

Resultats:

El temps mig d'obtenció el veredicte per l'algorisme de la Cerca Dicotòmica ha estat 19.69 segons i per l'algorisme de les Torres de Hanoi ha estat 13.61 segons. Cosa que fa que rebutgem H_0 (els dos temps d'execució són iguals) i per tant, si que podem observar diferència entre els temps d'obtenció dels veredictes en el Jutge.

Conclusió:

Amb les dades obtingudes no podem afirmar que l'algorisme de Cerca Dicotòmica obtingui abans el veredicte que el algorisme de les Torres de Hanoi. És per això que en la nostra mostra podem afirmar que la velocitat de correcció no depèn de la complexitat del algorisme. I això ho podem afirmar, observant el temps mig que tarda cada algorisme en ser corregit, amb un grau de confiança del 95%, un interval de confiança **IC = [5.658907, 6.517760]** i un p-valor de 2.2e-16. Per obtenir una idea general de com depenen els costos dels algorismes amb el temps per obtenir el veredicte, hauríem de utilitzar més d'un algorisme identificatiu per cada cost model d'algorisme.

Introducció

En varies assignatures hem fet servir aquesta web per saber si els nostres programes eren vàlids. Esperar uns segons, mai ens ha importat gaire, però ara volem saber si depenent de la complexitat del nostre algoritme haurem d'esperar més temps o menys per rebre el fantàstic verd o el decepcionant vermell.

És per això que triem dos algorismes coneguts, la Cerca Dicotòmica i les Torres de Hanoi els quals tenen uns costos ven diferenciats. Per calcular els costos de cada algorisme, hem de tenir clares unes certes premisses:

- Tots els algorismes consumeixen un temps, que depèn entre altres, de la talla de les seves entrades i de l'eficiència del algorisme sobre el qual s'executen. I es per això, que mesurarem el temps en funció de la talla de la entrada.

- Existeixen uns teoremes per tal de calcular els costos dels algorismes, en concret utilitzarem el Teorema de la resolució de recursives divisòries (*master theorem*), el qual ens permet calcular el cost d'algorismes recursius, per la Cerca Dicotòmica; i per les Torres de Hanoi utilitzarem el Teorema de resolució de recurrències subtractives.

El *Master theorem* ens diu que si tenim un algorisme recursiu de la forma:

$T(n) = a*T(n/b) + g(n)$ amb $b > 1$ i $g(n) = \Theta(n^k)$ i $k >= 0$.

On a = nombre de vegades que es repeteix el pas recursiu.

b = nombre en que dividim la talla en cada pas.

Podrem trobar $\alpha = \log_b (a)$

Tindrem:

$$T(n) = \begin{cases} \Theta(n^k) \text{ si } \alpha < k \\ \Theta(n^k \log_b(n)) \text{ si } \alpha = k \\ \Theta(n^\alpha) \text{ si } \alpha > k \end{cases}$$

El Teorema de resolució de recurrències subtractives ens diu que si tenim un algorisme recursiu de la forma:

T(n) = a*T(n-c) + g(n) amb c >= 1 i g(n) = $\Theta(n^k)$ i k >= 0.

On a = nombre de vegades que es repeteix el pas recursiu.

c = nombre en que reduïm la talla en cada pas.

Tindrem:

$$T(n) = \begin{cases} \Theta(n^k) \text{ si } a < 1 \\ \Theta(n^{k+1}) \text{ si } a = 1 \\ \Theta(a^{n/c}) \text{ si } a > 1 \end{cases}$$

Aplicant els Teoremes als nostres algorismes trobem:

Cerca Dicotòmica <- *Mester Theoreme*

T(n) = 1*T(n/2) + Θ(1) **α = 0**

Per tant el **T(n) = log_2(n)**

Torres de Hanoi <- Teorema de resolució de recurrències subtractives

T(n) = 2*T(n-1) + Θ(1)

Per tant el **T(n) = 2^n**

D'aquesta manera queden demostrats els costos dels algorismes en funció de les seves talles i això ens permet afirmar que tenen uns costos clarament diferenciats, on el cost de la Cerca Dicotòmica creix molt poc en funció de la talla de la entrada i les Torres de Hanoi, el seu cost, creix exponencialment en funció de la entrada de les dades. I a partir d'aquesta demostració i comprovació dels costos podrem inicialitzar el nostre estudi.

Objectiu:

Escollint aquest dos algorismes de costos clarament diferenciats com a mostra, un de logarítmic (Cerca Dicotòmica) i un d'exponencial (Torres de Hanoi). Ens plantegem l'objectiu de veure si influeix el cost d'aquets algorismes en l'obtenció del veredicte. En concret, si obtenim abans el veredicte de la Cerca Dicotòmica que el de Torres de Hanoi.

Material i mètodes

Per realitzar les proves em fet 30 enviaments amb cada codi a la web del Jutge (un total de 60 enviaments), en un ordre aleatori, per evitar pogués influenciar l'ordre d'enviament en la velocitat d'obtenció del veredicte. Tots els enviaments s'han fet des de els ordinadors de la FIB, per tal d'assegurar una connexió i un ample de banda similars. I tot els enviaments s'han fet entre les 9 i les 10 del matí.

Per obtenir les dades (temps) ens hem ajudat d'un cronòmetre, mesurant el temps que triguem des de que enviem l'arxiu fins veure el semàfor verd, en segons.

Per a l'estudi de les dades hem utilitzat el software estadístic R v.2.12.1, de la companyia *The R Foundation for Statistical Computing*.

A partir d'ara, per referir-nos al algorisme de Cerca Dicotòmica, parlarem de CD i per referir-nos al algorisme de Torres de Hanoi, parlarem de TH.

I. **Premisses**

· Normalitat per a cadascun de les variables
· Mostres aleatòries i independents
. Variàncies iguals

II. **Prova d'hipòtesis (bilateral):**

$$\begin{cases} H_0 : \mu_{DC} = \mu_{TH} \\ H_1 : \mu_{DC} \neq \mu_{TC} \end{cases}$$

III. **Càlcul de l'estadístic:**

Les dades són independents, per tat utilitzem una distribució T-student, ja que es compleix la premissa n_1 i n_2 >= 30 i o_1 i o_2 desconegudes.

μ	$H_0 : \mu_1 = \mu_2$	$\hat{t} = \frac{(\bar{y}_1 - \bar{y}_2)}{S\sqrt{1/n_1 + 1/n_2}}$ $S^2 = \frac{(n_1-1)S_1^2 + (n_2-1)S_2^2}{n_1+n_2-2}$	$Y_1, Y_2 \rightarrow N$ $\sigma_1 = \sigma_2$ m.a.s indep.	$\hat{t} \rightarrow t_{n1+n2-2}$	Rebutjar si $\lvert \hat{t} \rvert > t_{n1+n2-2,\ 1-\alpha/2}$ $\lvert \hat{t} \rvert > t_{n1+n2-2,0'975}$ amb α=5%

fragment extret dels apunts de la assignatura

y_1 -> estimador de μ_{CD}
y_2 -> estimador de μ_{TH}
S -> estimador de la desviació comuna de les mostres.

IV. **Estudi del p-valor (α = 0.05)**

Si el p-valor < α podem rebutjar la hipòtesis nul·la, amb una confiança del (1- α)%.

V. **Interval de confiança**

$$IC(\mu_T - \mu_O, 1-\alpha) = \left[\bar{x}_T - \bar{x}_O \mp t_{n_T+n_O-2,\ 0.975} \cdot s_p \sqrt{\frac{1}{n_T} + \frac{1}{n_O}} \right]$$

Resultats

Descriptiva

Les següents taules mostres un resum de les dades obtingudes per a cada algorisme. El nombre d'observació per cada un es: n = 30.

Cerca dicotòmica summary(CD)

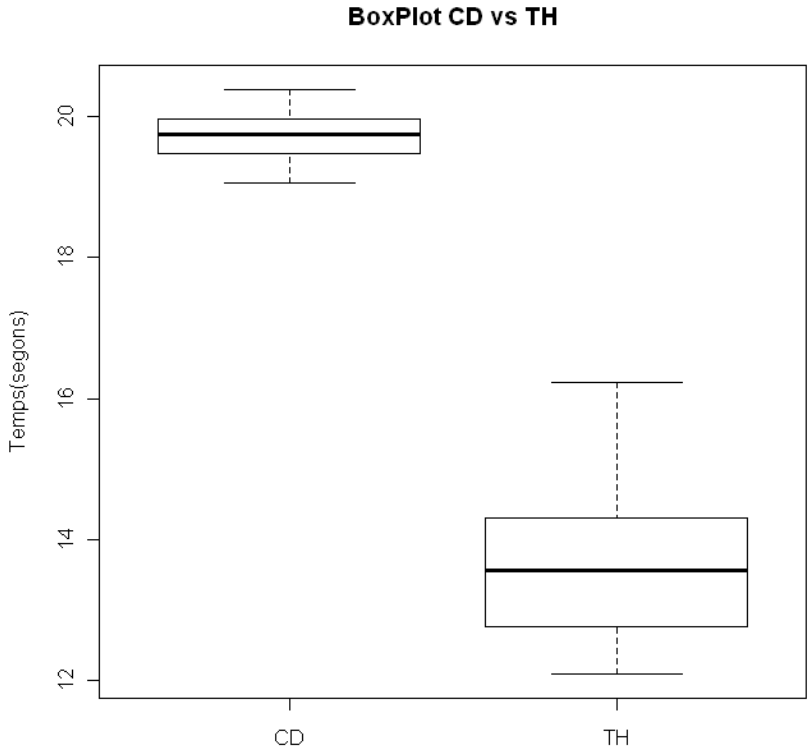
Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
19.05	19.49	19.75	19.69	19.95	20.39

Torres de Hanoi summary(TH)

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
12.09	12.77	13.57	13.61	14.29	16.23

Ambdues taules estan expressades en segons. Els càlculs son de la funció summary() del R

La següent figura veiem gràficament les dades obtingudes de les dues variables, mitjançant la funció box plot de R.




```
box(CD, TH, names = c("CD", "TH"), main = "BoxPlot CD vs TH", ylab = "Temps(segons)")
```

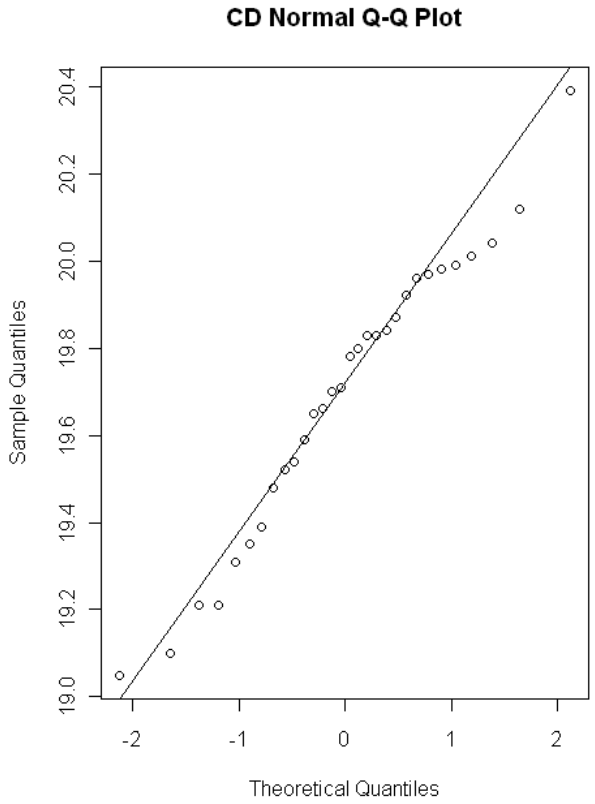
Pel gràfic podem suposar que les variàncies son diferents, per tant ho comprovem amb el var.test:

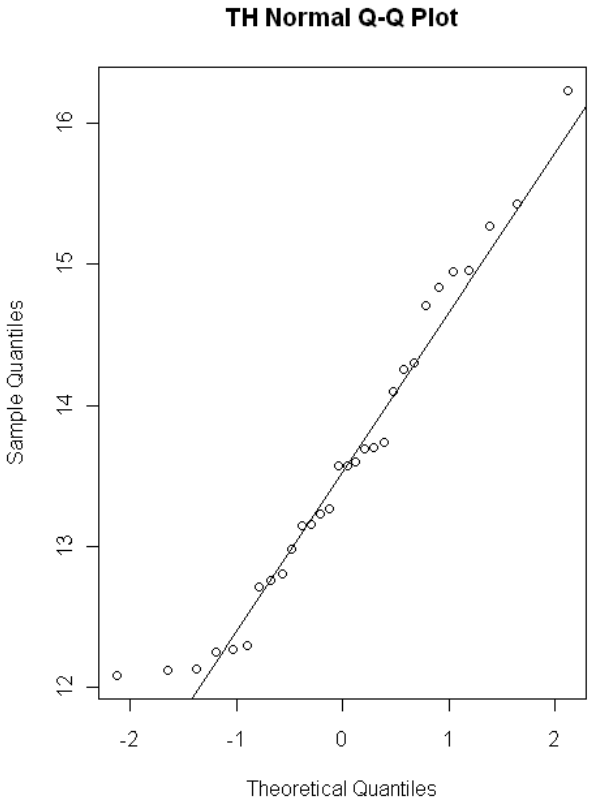
```
var.test(CD, TH)
```

```
        F test to compare two variances

data:  CD and TH
F = 0.0858, num df = 29, denom df = 29, p-value = 2.914e-09
alternative hypothesis: true ratio of variances is not equal to 1
95 percent confidence interval:
 0.04083579 0.18025666
sample estimates:
ratio of variances
         0.08579582
```

Per comprovar la normalitat de les dades utilitzem el qqnorm(), i com veiem que les dades es distribueixen al voltant de la recta, podem acceptar la normalitat.





Prova d'hipòtesi

Per provar d'hipòtesis, efectuem un t.test amb R, per poder trobar el p-valor i l' Interval de Confiança, cosa que ens farà rebutjar d'hipòtesis nul·la:

```
Welch Two Sample t-test

data:  CD and TH
t = 28.8147, df = 33.94, p-value < 2.2e-16
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 5.658907 6.517760
sample estimates:
mean of x mean of y
 19.69333  13.60500
```

t.test(x, y, alternative = "greater") en r

Trobem llavors que **p-value ≈ 0**, ja que un exponent de -16 el fa molt petit. El valor de l'estadístic **T** és **28.8147,** amb 58 graus de llibertat (30 observacions de dicotòmica + 30 observacions de Hanoi – 2). Amb un nivell de significació del 5% podem assegurar que el temps de correcció de l'algoritme de temps log2(n) (Cerca Dicotòmica) no es menor que el del algoritme 2^n (Torres de Hanoi) . L'estimació de la deferència es **6.08833**, l'extrem inferior del **IC = [5.658907, 6.517760]**.

Discussió

Després de fer el nostre estudi, veiem clarament que l'algorisme que l'algorisme de a Cerca Dicotòmica (amb un cost clarament inferior, log2(n)) té un temps d'obtenció del veredicte major en el jutge que l'algorisme de Torres de Hanoi (amb un cost clarament superior, 2^n). Podríem suposar que triga més un algorisme de cost inferior ja que el jutge executaria un major número de jocs de proves, que per un amb cost major, ja que això faria consumir més recursos al servidor i seria més fàcil que s'arribés a penjar.

Això ens porta a plantejar-nos, un possible estudi que tingui en compte tots els costos d'algorismes significatius i una quantitat d'execucions elevada per cada algorisme i a més, més d'un algorisme de cada tipus de cost. D'aquesta manera segurament podríem veure amb precisió, claredat i de manera general si el cost del algorisme és el que condiciona el temps per obtenir el veredicte. Però aquesta opció ara per ara, amb els coneixements que tenim de probabilitat i estadística no es possible, per falta de coneixement. A més, el jutge té una limitació de 40 enviaments per dia, cosa que faria que l'obtenció de les mostres s'allargués molt.

ANNEX I – Codis algorismes C++

Cerca Dicotòmica

```
#include <iostream>
#include <vector>

using namespace std;


int posicio(double x, vector<double>& v, int esq, int dre) {
  if(dre < esq) return -1;
  int mig = (esq+dre)/2;
  if(v[mig] < x) return posicio(x, v, mig+1, dre);
  if(v[mig] > x) return posicio(x, v, esq, mig-1);
  if(v[mig] == x) return mig;
}


int main() {
  int n;
  cin >> n;
  vector<double> v(n);
  for(int i = 0; i < n; ++i) cin >> v[i];
  double x;
  cin >> x;
  int pos = posicio(x, v, 0, n);
  cout << pos << endl;
}
```

Torres de Hanoi

```
#include <iostream>

using namespace std;
void Hanoi(int n, char from, char to, char aux) {
    if(n>0) {
        Hanoi(n-1, from, aux, to);
        cout << from << " => " << to << endl;
        Hanoi(n-1, aux, to, from);
    }
}

int main() {
     int n; cin >> n;
     Hanoi(n, 'A', 'C', 'B');
}
```

ANEX II – Script en R

```
# Llegir dades
CD <- read.table('clipboard')
TH <- read.table('clipboard')
CD<-CD$V1
TH<-TH$V1

#Obtenció dels valors representatius de les dades
summary(CD)
summary(TH)

#BoxPlot
box(CD, TH, names = c("CD", "TH"), main = "BoxPlot CD vs TH", ylab = "Temps(segons)")

# Comprovació de la igualtat de variàncies
var.test(CD, TH)

# Comprovació de la normalitat mitjançant gràfics
par(mfrow = c(1, 2))
qqnorm(CD, main = "CD Normal Q-Q Plot")
qqline(CD)
qqnorm(TH, main = "TH Normal Q-Q Plot")
qqline(TH)


#Obtenció dels valors significatius de les dades
t.test(CD, TH, var.equal = F)
```

ANEX III – Dades

Dades obtingudes de les 30 execucions de cada codi sobre el Jutge

CERCA DICOTÒMICA (segons)		**TORRES DE HANOI** (segons)	
19.99	19.96	13.27	12.09
19.83	19.05	13.57	12.71
19.65	19.97	13.74	12.25
19.21	20.04	13.69	12.76
19.31	19.71	13.70	12.12
19.48	19.83	13.60	15.27
19.54	20.12	13.57	14.26
19.52	19.92	13.23	14.10
19.35	19.87	13.16	16.23
19.78	19.21	12.98	14.71
19.1	19.98	13.15	14.95
19.7	19.84	12.27	14.96
19.8	19.59	12.81	14.30
20.39	19.39	12.30	15.43
19.66	20.01	12.13	14.84