CS 260 – Homework #1

Chris Kasper

January 14, 2018

Problem 1 (1.10 from text)

For each distinct pair (*i*, *j*) of functions determine if *fi* is dominated by *fj*, and if *fj* is dominated by *fi* . Show your reasoning. Assume *f* and *g* are positive functions on [0, ∞)

Solution

|  |  |
| --- | --- |
|  |  |

|  |  |
| --- | --- |
|  |  |

|  |  |
| --- | --- |
|  |  |

|  |  |
| --- | --- |
|  |  |

|  |  |
| --- | --- |
|  |  |

|  |  |
| --- | --- |
|  |  |

Problem 2 (1.12 from text)

Let **N** be the set of positive integers, and let **R** be the set of real numbers. For each of the procedures in Exercise 1.12, provide a function

*t*: **N** -> **R** such that the computing time function of the procedure is dominated by *t*. Make sure your function *t* is reasonably tight.

Solution

1. **procedure** *matmpy* (*n*: integer);
   1. **var** 
      * 1. *i*, *j*, *k*: integer; O(1)
        2. **begin** 
           1. **for** *i* := 1 **to** *n* **do** O(n)

**for** *j* := 1 **to** *n* **do begin**  O(n)

*C*[*i*, *j*] := O;  O(1)

**for** *k* := 1 **to** *n* **do** O(n)

*C*[*i*, *j*] := *C*[*i*, *j,*] + *A*[*i*, *k*] \* *B*[*k*, *j*] O(1)

**end**

**end**

1. **procedure** *mystery* (*n*: integer);
   * + 1. **var** 
          1. *i*, *j*, *k*: integer; O(1)
          2. **begin**
          3. **for** *i*:= 1 **to** *n*-1 **do** O(n)
          4. **for** *j*:= *i* + 1 **to** *n* **do**   O(n)
          5. **for** *k* := 1 **to** *j* **do** O(n)
          6. { some statement requiring *O*(1) time }   O(1)
          7. **end**
2. **procedure** *veryodd* ( *n*: integer );

**var**

*i*, *j*, *x*, *y*: integer; O(1)  
 **begin**

* + 1. **for** *i* := 1 **to** *n* **do** O(n)
  1. **if** *odd*(*i*) **then begin** O(1)

**for** *j* := *i* **to** *n* **do** O(n)

*x* := *x* + 1; O(1)

**for** *j* := 1 **to** *i* **do** O(n)

*y* := *y* + l O(1)

**end**

**end**

1. **function** *recursive* (*n*: integer ) : integer;

**begin**

**if** *n* <= 1 **then** O(1)

**return** (l) O(1)

**else**

**return** (*recursive*(*n*-1) + *recursive*(*n*-1)) ?

**end**

recursive(2) requires 2 additional recursive calls  
recursive(3) requires 6 additional recursive calls  
recursive(4) requires 14 additional recursive calls  
recursive(5) requires 30 additional recursive calls

mapping this as a function gives: