



UNIVERSITÀ DEGLI STUDI DELL'AQUILA  
**Algorithms for Distributed Systems: Mid-term Evaluation**  
 Tuesday, December 2nd, 2014 – Prof. Guido Proietti

Write your data $\implies$	Last name: ..... First name: ..... ID number: .....	Points
EXERCISE 1		
EXERCISE 2		
TOTAL		

**EXERCISE 1: Multiple-choice questions (20 points)**

**Remark:** Only one choice is correct. Use the enclosed grid to select your choice. A correct answer provides 3 points, while a wrong answer provides a  $-1$  penalization. The final result will be given by summing up all the obtained points (0 for a missing answer), by normalizing on a 20 base.

1. In un MPS *non uniforme*, i processori:
  - \*a) conoscono il numero totale di processori
  - b) sono tutti identici tra di loro
  - c) non conoscono il numero totale di processori
  - d) hanno identificativi distinti
2. In quale dei seguenti casi il problema dell'*elezione del leader* non è risolvibile:
  - a) anello asincrono, non anonimo e uniforme
  - b) anello sincrono, non anonimo e non uniforme
  - c) anello asincrono, non anonimo e non uniforme
  - \*d) anello sincrono, anonimo e non uniforme
3. L'algoritmo per l'*elezione del leader* in un anello asincrono con  $n$  processori, non anonimo e non uniforme con complessità di messaggi di  $O(n \log n)$ , genera durante la fase  $i$  al più un numero di messaggi pari a:
  - a)  $4i$
  - \*b)  $4n$
  - c)  $n$
  - d)  $\Theta(n \log n)$
4. L'algoritmo più efficiente per l'*elezione del leader* in un anello asincrono con  $n$  processori, non anonimo e non uniforme ha una complessità temporale:
  - a)  $\Theta(n)$
  - \*b) non è definita
  - c)  $\Theta(n^2)$
  - d)  $\Theta(n \log n)$
5. Sia dato un anello sincrono, non anonimo e non uniforme con 5 processori, con identificativo massimo pari a 10. Nel caso peggiore, l'algoritmo più efficiente per l'*elezione del leader* termina dopo un numero di round pari a:
  - a) 30
  - b) l'algoritmo non esiste
  - \*c) 35
  - d) 55
6. Consideriamo il primo round della versione sincrona dell'algoritmo di Prim su  $n$  processori. Nel caso peggiore, quanti messaggi verranno inviati in tale round?
  - a)  $n$
  - b)  $n - 1$
  - \*c)  $O(1)$
  - d) nessun messaggio
7. Consideriamo la prima fase della versione sincrona dell'algoritmo di Gallager, Humblet e Spira (GHS) su  $n$  processori. Nel caso peggiore, di quanti round può consistere tale fase?
  - a) exactly  $n$
  - b)  $O(1)$
  - \*c)  $\Theta(n)$
  - d)  $O(\log n)$
8. Durante l'esecuzione dell'algoritmo GHS, su un arco appartenente al minimo albero ricoprente transitano un numero di messaggi:
  - a)  $\Theta(\log n)$
  - b)  $O(1)$
  - \*c)  $O(\log n)$
  - d)  $\Theta(n \log n)$
9. Supponiamo di implementare l'algoritmo GHS in un anello asincrono di  $n$  processori. Il numero di messaggi che circolano nella rete è:
  - a)  $\Theta(n)$
  - b)  $O(1)$
  - \*c)  $O(n \log n)$
  - d)  $\Theta(m + n \log n)$
10. Durante l'esecuzione dell'algoritmo di GHS per la determinazione del minimo albero ricoprente di un MPS asincrono, non anonimo di  $n$  processori, il numero massimo di assorbimenti è:
  - a)  $n - 1$
  - b)  $O(1)$
  - \*c)  $n - 2$
  - d)  $\Theta(\log n)$
11. L'algoritmo randomizzato per trovare un *insieme indipendente massimale* su un grafo di  $n$  nodi e con grado massimo  $d = \Theta(\log n)$ , con alta probabilità termina entro un numero di round pari a:
  - \*a)  $O(\log^2 n)$
  - b)  $O(1)$
  - c)  $\Theta(n)$
  - d)  $O(\log n)$
12. Il numero di messaggi generato in una fase dall'algoritmo randomizzato per trovare un *insieme indipendente massimale* su un grafo di  $n$  nodi e con grado massimo  $d = O(1)$ , è pari a:
  - \*a)  $O(n)$
  - b)  $O(\log n)$
  - c)  $\Theta(n^2)$
  - d)  $O(d \log n)$
13. Sia dato un sistema sincrono di  $n$  processori, di cui al più  $f$  suscettibili di fallimenti benigni. Supponiamo che tutti i processori sani abbiano input  $x$ , mentre tra i processori non sani l'input minimo sia  $y < x$ . Quale tra i seguenti valori non è un output ammissibile dell'algoritmo di consenso consistente di  $f + 1$  round?
  - a)  $x$
  - b)  $y$
  - \*c)  $z < y$
  - d)  $y < z < x$
14. Sia dato un sistema sincrono di  $n$  processori, di cui al più  $n - 1$  suscettibili di fallimenti benigni. Nel caso *migliore*, quanti messaggi circolano nella rete per la risoluzione del problema del consenso consistente di  $f + 1$  round?
  - a) 0
  - \*b)  $n$
  - c)  $\Theta(n^2)$
  - d) 1
15. Sia dato un sistema sincrono di  $n$  processori, di cui al più  $f < n/4$  suscettibili di fallimenti bizantini. Nel caso migliore, qual è la complessità di messaggi dell'algoritmo *Phase King* per la risoluzione del problema del consenso?
  - a)  $\Theta(n)$
  - \*b)  $\Theta(n^2 f)$
  - c)  $\Theta(n^2)$
  - d)  $\Theta(n f)$
16. Sia dato un sistema sincrono di 13 processori, di cui al più 3 suscettibili di fallimenti bizantini. Qual è il numero minimo di messaggi ricevuti da un generico processore sano durante l'esecuzione di una fase dell'algoritmo *Phase King*?
  - a) 14
  - \*b) 11
  - c) 13
  - d) 0
17. Sia dato un sistema sincrono di  $n \geq 3f + 1$  processori, di cui al più  $f$  suscettibili di fallimenti bizantini, e supponiamo che gli input dei processori siano polinomiali in  $n$ . In termini di bit utilizzati, qual è la complessità di messaggi dell'algoritmo per la risoluzione del problema del consenso con albero esponenziale durante l'esecuzione del  $k$ -esimo round?
  - \*a)  $\Theta(n^2 \log n^k)$
  - b)  $\Theta(n^2 \log k)$
  - c)  $\Theta(\log n^k)$
  - d)  $\Theta(n^k)$
18. Nell'algoritmo del *fornaio* per il problema della *mutua esclusione* in un sistema a memoria condivisa di  $n$  processori, un processore  $p_i$  che si trova nella *entry section*, prima di accedere alla *critical section* può essere preceduto da un fissato processore  $p_j, j > i$ , un numero di volte pari a:
  - a) illimitato
  - b)  $n - 1$
  - c)  $k$ , con  $k > 1$  fissato
  - \*d) 1
19. Nell'algoritmo del *fornaio* per il problema della *mutua esclusione* in un sistema a memoria condivisa di  $n$  processori, un processore  $p_i$  che si trova nella *entry section* ed ha già ritirato il suo ticket, prima di accedere alla *critical section* può essere preceduto da un numero massimo di processori (escluso quello correntemente nella *critical section*) pari a:
  - a) illimitato
  - \*b)  $n - 2$
  - c)  $k$ , con  $k$  costante
  - d) 1

20. Nell'algoritmo del *torneo* per il problema della *mutua esclusione* in un sistema a memoria condivisa di  $n$  processori, un processore  $p_i$  che si trova nella *entry section*, prima di accedere alla *critical section* può essere preceduto da un fissato processore  $p_{i+1}$  un numero di volte pari a:  
 \*a) illimitato    b)  $n - 1$     c)  $k$ , con  $k > 1$  fissato    d) 1

**Answer Grid**

	Question																			
Choice	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
a																				
b																				
c																				
d																				

**EXERCISE 2: Algorithm (10 points: 5 for the correctness, 3 for the efficiency, and 2 for the analysis)**

Realizzare ed analizzare un algoritmo distribuito che, preso in input un sistema sincrono di 4 processori sani  $p_1, p_2, p_3, p_4$  mutuamente connessi e con input  $x_1, x_2, x_3, x_4$ , risolva il problema del consenso nell'ipotesi che al più un arco del sistema fallisca in modo permanente.