Lista zadań. Nr 1. 16 lutego 2005

Algorytmy i Struktury Danych

IIUWr. II rok informatyki.

1. (0 pkt)Przeczytaj notatkę do pierwzego wykładu.
2. Skonstruuj programy dla maszyny RAM:
   (a) (1pkt) liczący $n!$ dla danego n,
   (b) (1pkt) czytający n liczb naturalnych zakończony liczbą zero (n nie jest dane), a następnie wypisujący te liczby w porządku niemalejącym,

   Zbadaj złożoność czasową i pamięciową napisanych programów przy jednorodnym i logarytmicznym kryterium kosztów. Ustal własną miarę "rozmiaru" danych.
3. (1pkt) Udowodnij, że algorytm mnożenia liczb *"po rosyjsku"* jest poprawny. Jaka jest jego złożoność czasowa i pamięciowa przy:
   - jednorodnym kryterium kosztów,
   - logarytmicznym kryterium kosztów?
4. (2pkt) Czy algorytm "macierzowy" obliczania n-tej liczby Fibonacciego można uogólnić na inne ciągi, w których kolejne elementy definiowane są liniową kombinacją skończonej liczby elementów wcześniejszych? Czy zadanie skomplikuje się, jeśli dopuścimy występowanie stałego składnika w kombinacji liniowej?
5. (1pkt) Przypomnij algorytm sortowania bąbelkowego. Zapisz go w notacji zbliżonej do tej, której używaliśmy na wykładzie. Porównaj go z algorytmami *InsertSort* i *SelectSort* stosując podane na wykładzie kryteria.

Zadania dodatkowe dla grupy zaawansowanej

1. (2pkt) Rozważ następujący problem:

   Problem.[1]
   dane: $n, m \in \mathcal{N}$
   wynik: wartość współczynnika przy x^2 (wzięta modulo m) wielomianu $\underbrace{(...((x-2)^2-2)^2...-2)^2}_{n \text{ razy}}$

   Czy widzisz zastosowanie metody użytej w szybkim algorytmie obliczania n-tej liczby Fibonacciego do rozwiązania tego problemu?

Zadania nadobowiązkowe - nie będą rozwiązywane w czasie ćwiczeń

1. (1pkt) Co stałoby się z mocą obliczeniową maszyny RAM gdyby instrukcje ADD i MULT zostały usunięte z repertuaru instrukcji? Jak zmieniłby się koszt obliczeń?
2. (1pkt) Pokaż, że dla każdego programu maszyny RAM istnieje równoważny program maszyny RAM (tj. taki, który dla tych samych danych produkuje te same wyniki) używający nie więcej niż 2^{14} komórek pamięci.

[1] Zadanie zaczerpnięte ze Sparingu w Programowaniu Zespołowym - Poznań 22.01.2005

3. (0pkt) Przypomnij sobie notację asymptotyczną dla rzędów funkcji: O, Ω, Θ.

4. (1pkt) Jaka jest najmniejsza wartość n, dla której algorytm o złożoności $100n^2$ działa (na tej samej maszynie) szybciej od algorytmu o złożoności 2^n?

5. (1pkt) Dla każdej funkcji $f(n)$ i czasu t w poniższej tabelce, określ największy rozmiar n danych, dla których algorytm wykona obliczenia w czasie t. Zakładamy, że algorytm rozwiązujący problem potrzebuje $f(n)$ mikrosekund dla danych rozmiaru n.

	1 sekunda	1 minuta	1 godzina	1 dzień	1 miesiąc	1 rok	1 wiek
$\log n$							
$\sqrt{n}$							
n							
$n \log n$							
n^2							
n^3							
2^n							
$n!$							

O ile większe zadania można by rozwiązywać na komputerze 1000 razy szybszym (tj. takim, na którym algorytm potrzebowałby $f(n)$ nanosekund dla danych rozmiaru n)?

6. (1pkt) Napisz w C lub Pascalu funkcję implementującą podany na wykładzie algorytm, który oblicza n-tą liczbę Fibonacciego (modulo stała) w czasie $O(\log n)$.

Krzysztof Loryś