Zadania przygtowawcze 2

Piotr Chrząstowski-Wachtel

1. Napisz funkcję

   ```
   bool Iloczyny6(int T[], int n, int k),
   ```

   która przyjmie wartość "true", wtedy i tylko wtedy gdy iloczyn liczb zapisanych w każdym segmencie tablicy T długości k jest podzielny przez 6 dla k,n>0.

2. Tablica A zawiera albo liczby dodatnie, albo wartości -1, przy czym wszystkie wartości -1 występują w jednym spójnym segmencie. Ponadto dla dowolnych i, j takich że $0 \leq i \leq j$ <n zachodzi $(A[i] > 0 \wedge A[j] > 0) \rightarrow A[i] \leq A[j]$. Napisz funkcję `int znajdz(int A[], int n, int x)` , która znajdzie w tablicy A indeks zadanej liczby dodatniej x, o ile ona się tam znajduje. Jeśli liczby x nie ma w A, to wartością funkcji powinno być -1.

3. Napisz funkcję `bool takiesame(int A[], int N)`, która sprawdzi, czy z zapisanego w tablicy ciągu 2N liczb całkowitych można wybrać N-elementowy podciąg tak, by ciągi elementów wybranych i niewybranych były takie same.

4. Dla tablicy A oraz liczby całkowitej k powiemy, że segment `A[d..g]` jest k-płaski, jeśli dla każdych i, j takich, że $d \leq i, j \leq g$ zachodzi nierówność $A[i] - A[j] \leq k$. Napisz funkcję `int kaplaski(int A[], int n, int k)`, która dla tablicy A długości n wyznaczy długość najdłuższego k-płaskiego segmentu.

5. Niech $T = \{0, ..., n-1\}$. Funkcja $f : T \rightarrow T$ jest niemalejąca, a jej wartości są zapisane w tablicy `int F[n]` tak, że $f(k) = F[k]$ dla $k = 0, \ldots, n-1$. Definiujemy funkcję $gf : T \rightarrow T$ w następujący sposób: $gf(k) = |\{a \in T : f(a) = k\}|$, gdzie przez $|X|$ oznaczamy liczność (moc) zbioru X. Napisz fragment programu, który zmiennej *rekord* nada wartość równą maksimum spośród wartości przeciwobrazów funkcji gf z jednoelementowych podzbiorów zbioru T, czyli $max(|\{gf^{-1}(\{k\})| : k = 1, \ldots, n\})$.

6. Przez segment słowa $v \in A^*$ składający się z samych liter x nazwiemy takie słowo $w = x^k$ dla pewnego $x \in A$ oraz $k \in N_+$, że $v = w' w w''$ dla pewnych $w', w'' \in A^*$. Segment taki jest maksymalny, jeśli ani w' nie kończy się na x, ani w'' nie zaczyna się od x. Napisz gramatykę generującą wszystkie słowa v nad alfabetem $\{a, b, c\}$ o tej właściwości, że każdy maksymalny segment złożony z samych liter b w słowie v musi być bezpośrednio poprzedzony przez dłuższy segment złożony z samych liter a.

7. Dane są dwie funkcje $f, g : \{0,1\}^* \rightarrow Z$ określone w następujący sposób: $f(\varepsilon) = 0, f(0w) = g(w) + 1, f(1w) = g(w) - 1, g(\varepsilon) = 0, g(0w) = f(w) - 1, g(1w) = f(w) + 1$ dla $w \in \{0,1\}^*$. Zdefiniujmy język $L = \{w \in \{0,1\}^* : f(w) = 0\}$. Napisz funkcję `bool JestL(int T[], int n)`, która sprawdzi, czy słowo zapisane w tablicy T należy do języka L.

8. Dana jest następująca funkcja:

```
int coto(int n)
{ int i,x,y;
  x=0; y=1; i=2;
  while (i<=n)
   { x+=y;
     y=x-y;
     i+=2;
   }
  if (i%2) return y;
  else return x;
}
```

Określ dla każdego naturalnego n, co jest wartością funkcji `coto(n)` i udowodnij to.

9. Dana jest następująca funkcja:

```
int coto2(int n)
{ int x,y;
  x=0; y=1;
  while (y<=n)
   { x-=y;
     y-=x;
   }
  if (i%2) return y;
  else return x;
}
```

Określ dla każdego naturalnego n, co jest wartością funkcji `coto(n)` i udowodnij to.

10. Dany jest ciąg prostokątów $P_1, \ldots, P_n$. Tablice `int A[],B[]` długości $n > 0$ reprezentują długości boków tych prostokątów tak, że dla każdego $i = 0, \ldots, n-1$ para `(A[i],B[i])` określa długości boków prostokąta P_{i+1}. Ciąg tych prostokątów ma tę szczególną własność, że tablica `A` jest posortowana niemalejąco, a tablica `B` nierosnąco. Napisz funkcję `float int minprzek(int A[], int B[], int n)`, która wyznaczy indeks prostokąta o najkrótszej przekątnej.

11. Napisz funkcję `float mindist(float A[], float B[], int n)`, która obliczy wartość $min(|A[k] - B[k]| : 0 \leq k < n)$, przy założeniu, że tablica `A` jest posortowana rosnąco, a `B` malejąco.

12. Podaj gramatykę języka $L = \{w \in \{a,b\}^* : \#(a,w) = 2\#(b,w)\}$.

13. W tablicy `float A[]` długości n jest zapisana wysokość pokrywy śnieżnej na dachu pewnego budynku w dniach 1..n i wiemy, że jest to ciąg słabo bitoniczny (najpierw niemalejący, potem nierosnący). Każdy centymetr świeżego śniegu (z danego dnia) waży

nw, zaś centymetr śniegu z dnia poprzedniego jest c razy cięższy niż w dniu poprzednim. Kolejne warstwy śniegu układają się na poprzednich; jeśli śnieg topnieje, to dzieje sie to w kolejności odwrotnej, czyli najpierw topnieje śnieg z wczoraj, potem zprzedwczoraj itd. Napisz funkcję `bool zaciezko(int A[], float c, float r, float nw)`, która sprawdzi, czy był taki dzień, w którym ciężar śniegu na dachu przekroczył wartość krytyczną r.

14. Niech A będzie tablicą liczb całkowitych o dodatniej parzystej dłogości. Napisz funkcję `bool kantybiton(int A[] int n`, sprawdzającą, czy istnieje taka liczba k z przedziału $0, \ldots, n-2$, że tablica B, równa tablicy A przesuniętej cyklicznie o k elementów w prawo, jest antybitoniczna, czyli od początku do pewnego miejsca malejąca i od tego samego miejsca do końca rosnąca.

15. Rozważamy funkcję $f : \{0, \ldots, n-1\} \to \{0, \ldots, n-1\}$. Dla każdego dodatniego k określamy $f^k(x) = f(x)$ dla k=1 oraz $f^k(x) = f^{k-1}(f(x))$ dla $k > 1$. Powiemy, że y jest zależne od x jeśli istnieje $k > 0$ takie, że $y = f^k(x)$. O funkcji f wiemy tyle, że wyznaczona przez nią relacja zależności jest antysymetryczna. Dwa elementy x i y są niezależne, jeśli ani x nie jest zależne od y ani y od x. Zbiór X nazwiemy niezależnym, jeśli każde dwa jego elementy są niezależne. Funkcję f definiujemy za pomocą tablicy długości n o wartościach ze zbioru $0, \ldots, n-1$. Napisz funkcję `int maxniezalezne(int f[], int n)`, która wyznaczy liczność najliczniejszego niezależnego podzbioru zbioru $\{0, ..., n-1\}$ wyznaczonego przez funkcję f.

16. Tablica A zawiera same jedynki i zera. Napisz funkcję `int maxrot(int A[], int n)`, która wyznaczy takie k, żeby jedynki i zera w tablicy powstałej przez przesunięcie A o k pozycji w prawo ułożyły się w możliwie dużą liczbę w zapisie dwójkowym. *Zrobienie tegow czasie liniowym jest możliwe, ale wykracza poza zakres materiału tego semestru*

17. Dane są dwie liczby rzeczywiste. Każda z nich jest reprezentowana w dwóch zerojedynkowych tablicach: w tablicy n-elementowej zapisana jest cecha, a w m-elementowej mantysa. Napisz funkcję `bool wieksza(int C1[], int M1[], int C2[], int M2[], int n, int m)`, która sprawdzi, czy liczba zapisana w systemie zmiennopozycyjnym w kodzie uzupełnieniowym o cesze C1 i mantysie M1 jest większa niż liczba o cesze C2 i mantysie M2. Zakładamy, że minimalna cecha reprezentuje 0 i że nie ma ukrytego bitu jednej drugiej.

18. Dla typów, jak w zadaniu poprzednim, wygeneruj kolejną co do wielkości liczbę rzeczywistą do danej. `int kolejna(int C[], int M[], int n, int m)`. Jeśli mieliśmy największą możliwą liczbę w tablicach C i M, to zostaw tablice w spokoju i niczego nie zmieniaj.

19. Oblicz błąd względny wykonania działania $\frac{2}{7} + \frac{3}{11}$ w systemie zmiennopozycyjnym z wykładu.