Филиппова Анастасия

Домашнее задание 2

(1) Найдем верхнюю оценку:
$\sum \sqrt{i^3+2i+5} < n \cdot \sqrt{n^3+2n+5} \Rightarrow \sum \sqrt{i^3+2i+5} = O(n^{\frac{5}{2}})$
Найдем нижнюю оценку:
Для этого возьмем верние $\frac{n}{2}$ членов.Очевидно, что:
$\sum \sqrt{i^3+2i+5} > \frac{n}{2} \cdot \sqrt{\frac{n^3}{8}+n+5} \Rightarrow \sum \sqrt{i^3+2i+5} = \Omega(n^{\frac{5}{2}})$
Следовательно, $\sum \sqrt{i^3+2i+5} = \Theta(n^{\frac{5}{2}})$

(2) $f(n) = (3+o(1))^n + \Theta(n^{100})$
Верно ли, что: $\log f(n) = \Theta(n)$?
$f(n) > (3^n) + \Theta(n^{100}) < 3^n$
Так как $\forall\ n \in N\ f(n) > 0 \Rightarrow \log f(n) > n \cdot \log 3$, тогда $\log f(n) = \Omega(n)$
Пусть $o(1) = g(n)$,$\Theta(n^{100}) = h(n)$, получаем:
$\forall\ c > 0 \hookrightarrow g(n) < c$
$\exists c_1, c_2 > 0, n_0 \in N : \forall\ n \geqslant n_0 \hookrightarrow c_1 \cdot n^{100} \leqslant h(n) \leqslant c_2 \cdot n^{100}$
$f(n) < (3+c)^n + c_2 \cdot n^{100}$
Очевидно, что $3^n = \Omega(n^{100})$, тогда:
$\exists\ c' > 0, N \in N : \forall\ n \geqslant N \hookrightarrow (3+c)^n \geqslant c' \cdot n^{100} \Rightarrow$
$\exists\ C > 0 (C = 1/c') : f(n) < C(3+c)^n \Rightarrow \log f(n) = O(n)$
Следовательно, $\log f(n) = \Theta(n)$

(3) Пусть $b = bound$, опишим внутренний цикл:
$\sum\limits_{i=0}^{b-1} \frac{i}{2} + b\log n) = \frac{b(b-1)}{4} + b\log n = \frac{b^2-b}{4} + b\log n$
Внешний цикл:
$\sum\limits_{b=1}^{k} \frac{b^2-b}{4} + b\log n, \sqrt{n} > k > \sqrt{n}-1$, тогда:$\sum\limits_{b=1}^{k} \frac{b^2-b}{4} + b\log n = \Theta(n \cdot n^2) = \Theta(n^{\frac{3}{2}})$

(4) (a) $238x + 385y = 133$ $(238, 385) = (238, 147) = (147, 91) = (91, 56) = (56, 35) = (35, 21) = 7$
$7|133 \Rightarrow$ уравнение имеет целые решения.
Разделим обе части уравнения на 7:
$34x + 55y = 19$,частное решение:
$34x_0 + 55y_0 = 1$
$34 \begin{pmatrix} 1 \\ 0 \end{pmatrix}\ 55 \begin{pmatrix} 0 \\ 1 \end{pmatrix}$
$21 \begin{pmatrix} -1 \\ 1 \end{pmatrix}\ 34 \begin{pmatrix} 1 \\ 0 \end{pmatrix}$
$8 \begin{pmatrix} -3 \\ 2 \end{pmatrix}\ 13 \begin{pmatrix} 2 \\ -1 \end{pmatrix}$
$5 \begin{pmatrix} 5 \\ -3 \end{pmatrix}\ 8 \begin{pmatrix} -3 \\ 2 \end{pmatrix}$
$3 \begin{pmatrix} -8 \\ 5 \end{pmatrix}\ 5 \begin{pmatrix} 5 \\ -3 \end{pmatrix}$
$2 \begin{pmatrix} 13 \\ -8 \end{pmatrix}\ 3 \begin{pmatrix} -8 \\ 5 \end{pmatrix}$
$1 \begin{pmatrix} -21 \\ 13 \end{pmatrix}\ 2 \begin{pmatrix} 13 \\ -8 \end{pmatrix}$

Получили частное решение:
$x_0 = 19 \cdot (-21) = -399$
$y_0 = 19 \cdot 13 = 247$
Общее решение:
$x = -399 + k \cdot 55$
$y = 247 + k \cdot 34,\ k \in Z$

(b) $143x + 121y = 52$
$(143, 121) = 11$
$(52, 11) = 1 \Rightarrow$ решений нет.

(5) (с семинара)

$3^{11}\ mod\ 102$

$1)(3,5,102), return\ 3\cdot(3^5\ mod\ 102)^2$

$2)(3,2,102), return\ 3\cdot(3^2\ mod\ 102)^2$

$3)(3,1,102), return\ 3\cdot(3^1\ mod\ 102)^2$

$4)(3,0,102), return\ 1$

Получаем:$4)1, 3)9\cdot 3 = 27, 2)81\cdot 3\ mod\ 102 = 29, 1)(29\cdot 29\ mod\ 102)\cdot 3\ mod\ 102 = 75$

(7) (с семинара)

Псевдо - код:

Multiply(x, y) **if** *($y == 0$)* **then**
| return 0; $z = Multiply(x; \frac{y}{2})$;
end
if *($y\ mod\ 2 == 0$)* **then**
| return $2z$;
end
else
| return $x + 2z$
end

Данный алгоритм корректен, так как сделует правилу:

при $y = 0$ возвращает 0;

при $y \neq 0$: $2\cdot x\frac{y}{2}$, при y четном, $2\cdot x\frac{y}{2} + x$, при y нечетном.

Сложность алгоритма:

если запись числа содержит n битов, то для завершения алгоритма понадобится n рекурсивных вызовов(так как при каждом таком вызове длина числа уменьшается на 1).В каждом рекрсивном вызове происходит $O(n)$ битовых операций.Получаем сложность $O(n^2)$.

(5) (домашнее задание)

Задача: найти такие $q, r : x = yq + r$.

Divide(x, y) **if** *($x == 0$)* **then**
| return $(0, 0)$;
end
$(q, r) = Multiply((\frac{x}{2}); y)$
q = 2q;
r = 2r;
if *($x\ mod\ 2 \neq 0$)* **then**
| $r = r + 1$;
end
if *($r \geqslant y$)* **then**
| $r = r - y$;
| q = q +1;
end
$return(q, r)$;

Очевидно, что данный алгоритм корректен, потому что подчиняется следующему правилу:

при $x = 0$ возвращает $(0, 0)$,

при $x \neq 0$:

если x - четное и $r < y : x = 2[x/2] + 2r$;

если x - четное и $r >= y : x = 2[x/2] + 1 + r - y$;

если x - не четное и $r < y : x = 2[x/2] + 2r + 1$;

если x - не четное и $r > y : x = 2[x/2] + 2r + y + 1$.

Сложность алгоритма:

Аналогична алгоритму умножения: если число x записывается с помощью n битов, то для завершения алгоритма требуется n рекурсивных вызовов.При каждом рекурсивном вызове совершается $O(n)$ битовых операций.Получаем $O(n)$ битовых операций для n-битовых входов $\Rightarrow$ время работы алгоритма $O(n^2)$.