W6. Rekurencje

Algorytmy często przywołują same siebie jako podprocedury (rekurencja, rekursja, ~~iteracja~~)

<u>Pr1</u> Przeszukiwanie binarne (binary search) uporządkowanej listy w celu lokalizacji elementu. Polega na podzieleniu listy na pół, stwierdzeniu, do której połowy należy szukany element, i zastosowaniu całej procedury do tej połowy.

$f(n) = \#$ porównań wykonanych przez algorytm

$\boxed{f(n) = f\left(\frac{n}{2}\right) + 2}$ ← wybór połówki listy ← sprawdz., czy > 1 el. na liście

Przeliczanie też często ma charakter rekurencyjny.

<u>Pr2</u> $a_n = \#$ permutacji rzędu n, np. zbioru $\{1, \dots, n\}$

$a_0 = 1,\ a_1 = 1,\ a_2 = 2$, $\boxed{a_n = n a_{n-1},\ n \geq 1}$

bo n można „wcisnąć" na n sposobów w perm. rzędu $n-1$.

<u>Hip</u> $a_n = n!$ <u>Dow. ind.</u> $a_n = n a_{n-1} = n \cdot (n-1)! = n!$ □

Prz 3 Na ile obszarów dzieli płaszczyznę n prostych w pozycji ogólnej?

(żadne 2 nie są równoległe, żadne 3 nie przecinają się w 1p.)

$a_0 = 1$, $a_1 = 2$, $a_2 = 4$, $a_3 = 7$, $a_4 = 11$

$$\boxed{a_n = a_{n-1} + n, \quad n \geq 1}$$

$$a_n = a_{n-1} + n = a_{n-2} + (n-1) + n = \ldots = a_0 + 1 + 2 + \ldots + n$$

Hip $a_n = \binom{n+1}{2} + 1$ Dow. ind. $a_n = \binom{n}{2} + 1 + n$ $\square$

Prz 4 „Wieża w Hanoi" (Lucas, 2 poł. XIX w.)

3 paliki: 2 puste, a na 1 włożono $n = 64$ pierścieni różnej wielkości, od największego (na spodzie), do najmn. Mnisi buddyjscy mają przenieść całą wieżę na inny palik, po 1 pierścieniu w każdym ruchu, nigdy nie kładąc większego na mniejszy. Jeśli każdy ruch trwa 1 sek., to kiedy nastąpi koniec świata?

Rozw: żeby ruszyć najw. pierścień P_n, trzeba przenieść całą wieżę $P_1, \ldots, P_{n-1}$ na inny palik, potem P_n na 3-ci palik, i ponownie $P_1, \ldots, P_{n-1}$ na P_n. Oznaczmy przez a_n minimalną # ruchów.

$a_0 = 0$, $a_1 = 1$, $a_2 = 3$, $a_3 = 7$, ...

$$\boxed{a_n = 2a_{n-1} + 1, \ n \geq 1}$$

Hip $a_n = 2^n - 1$ Dow. ind $a_n = 2(2^{n-1} - 1) + 1 = 2^n - 1$ □

$2^{64} - 1 \approx 18{,}5 \cdot 10^{18}$ s. ≈ 500 mld lat

Pr5. Liczby Fibonacciego (Leonardo di Pisa, XIII w.)

Nowonarodzona para królików (różnej płci) zostaje umieszczona na wyspie. Każda para rodzeństwa, gdy osiągnie 2 miesiące, wydaje co miesiąc na świat nową parę. Ile par królików będzie na wyspie po n miesiącach, jeśli króliki są nieśmiertelne?

Rozw $f_0=0,\ f_1=1,\ f_2=1,\ f_3=2,\ f_4=3, \ldots$

$$\boxed{f_n = f_{n-1} + f_{n-2}\ ,\ n \geq 2}$$

$$f_n := x^n \Rightarrow x^2 = x+1 \Rightarrow x_{1,2} = \frac{1 \pm \sqrt{5}}{2}$$

$$f_n = a\left(\frac{1+\sqrt{5}}{2}\right)^n + b\left(\frac{1-\sqrt{5}}{2}\right)^n \xRightarrow[n=1]{n=0} a = \frac{1}{\sqrt{5}},\ b = \frac{-1}{\sqrt{5}}$$

$$\Rightarrow f_n = \frac{1}{\sqrt{5}}\left(\frac{1+\sqrt{5}}{2}\right)^n - \frac{1}{\sqrt{5}}\left(\frac{1-\sqrt{5}}{2}\right)^n \quad \square$$

UWAGA! Czasem przyjmuje się $f_0=1,\ f_1=1,\ f_2=2, \ldots$
i wtedy $f_n = \frac{1}{\sqrt{5}}\left(\frac{1+\sqrt{5}}{2}\right)^{n+1} - \frac{1}{\sqrt{5}}\left(\frac{1-\sqrt{5}}{2}\right)^{n+1}$.

Prz 1 (c.d.) Jak rozwiązać $\boxed{f(n) = f\left(\frac{n}{2}\right) + 2}$, $f(1)=0$?

Załóżmy, że $n = 2^k$, $k \geq 0$.

$$f(n) = f(2^k) = f(2^{k-1}) + 2 = f(2^{k-2}) + 2 + 2 = f(2^{k-2}) + 4 =$$
$$= \ldots = f(2^0) + 2k = 2k = 2\log_2 n \quad \square$$

Ten typ rekurencji nosi nazwę „divide-and-conquer" i jest b. ważny w teorii algorytmów.

Prz6 Znajdowanie maksimum w ciągu $a_1, \dots, a_n$.

Alg: dzielimy na połowy, w nich znajdujemy max., porównujemy je ze sobą.

$$\boxed{f(n) = 2f\left(\frac{n}{2}\right) + 1}, \quad f(1) = 0, \quad n = 2^k, \ k \geq 0$$

$$f(n) = \dots = 2^k f(1) + \sum_{j=0}^{k-1} 2^j = 0 + 2^k - 1 = n - 1 \quad \square$$

Prz7 Merge Sort – sortowanie n-el. listy przez dzielenie na pół, sortowanie połówek osobno, i łączenie.

$$\boxed{f(n) = 2f\left(\frac{n}{2}\right) + n} \quad f(1) = 0, \quad n = 2^k, \ k \geq 0$$

Rozw: $a_k = f(2^k)$, $a_k = 2a_{k-1} + 2^k$, $a_0 = 0$

$$a_k = 2\left(2a_{k-2} + 2^{k-1}\right) + 2^k = 2^2 a_{k-2} + 2 \cdot 2^k$$

$$= \dots = k 2^k$$

$$\Rightarrow f(n) = n \log_2 n \quad \square$$