Implementacja listy na tablicy

S. Płaneta

26 marca 2015

1 Zadanie

Zadaniem było stworzenie utworzenie listy na tablicy, a następnie porównanie dwóch metod powiększania jej rozmiaru - zwiększanie rozmiaru o 1, oraz zwiększanie rozmiaru dwukrotnie.

Obliczanie złożoności obliczeniowej:
Do obliczenia złożoności obliczeniowej przyjęto założenia, że aktualny rozmiar tablicy wynosi rozm, oraz tablica jest całkowicie wypełniona. Pominięto również niektóre mniej znaczące operacje, które nie miałyby wpływu na asymptotyczną złożoność obliczeniową.

Dodanie n elementów n koniec listy, zwiekszając rozmiar tablicy o 1 przy dodawaniu każdego elementu.

Algorytm Zwiekszanie-o-1(tab,rozm,n)	Ilosc operacji
for i←1 **to** n **do**	n
for j←0 **to** rozm-1 **do**	rozm+(rozm+1)+...+(rozm+n-1)
nowatab[j]←tab[j]	rozm+(rozm+1)+...+(rozm+n-1)
tab←nowatab	n
rozm++	n
tab[rozm-1]←nowyelement	2n
return tab	1

Suma: $2*n(n+2rozm-1)/2+5n+1 \approx n(n+2rozm-1)$, dlatego algorytm działa w czasie $O(n^2)$

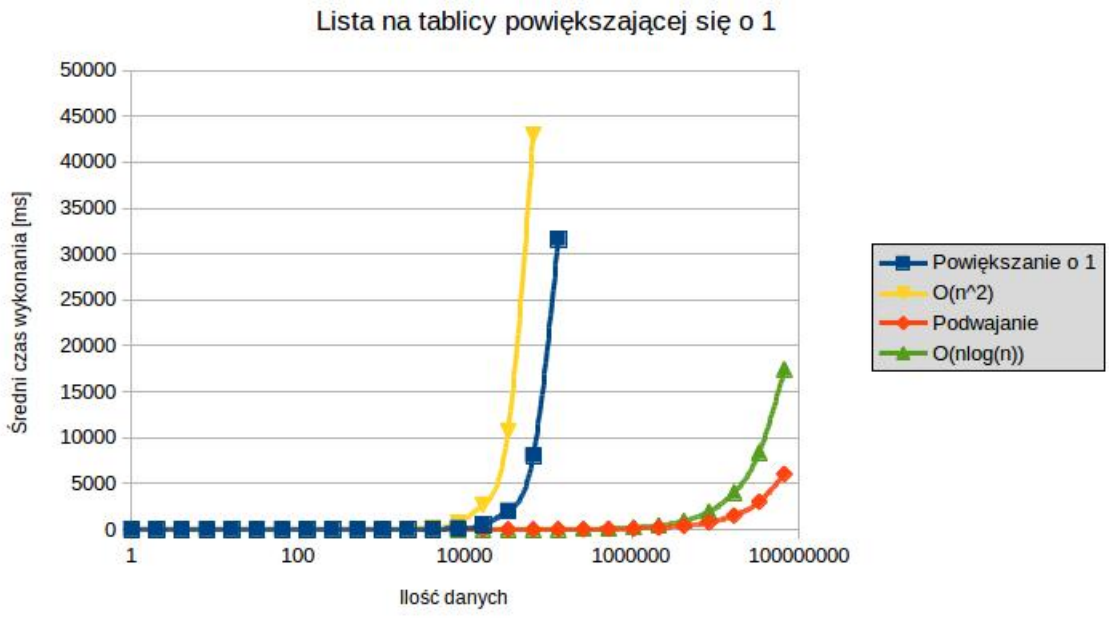
Dodanie n elementów n koniec listy, zwiekszając rozmiar tablicy dwukrotnie w wypadku gdy zostanie przekroczony rozmiar tablicy. Ile - liczba elementów aktualnie znajdujących się w tablicy.

Algorytm Zwiekszanie-2x(tab,rozm,ile,n)	Ilosc operacji
for i←1 **to** n **do**	n
if ile<rozm **then**	n
tab[ile]←nowyelement	$2(n - \lceil \log_2(rozm+n) \rceil)$
ile++	$(n - \lceil \log_2(rozm+n) \rceil)$
else	
for j←0 **to** rozm-1 **do**	$\sim (rozm+n) * \lceil \log_2(rozm+n) \rceil$
nowatab[j]←tab[j]	$\sim (rozm+n) * \lceil \log_2(rozm+n) \rceil$
tab←nowatab	$\lceil \log_2(rozm+n) \rceil$
tab[rozm]←nowyelement	$2\lceil \log_2(rozm+n) \rceil$
ile++	$\lceil \log_2(rozm+n) \rceil$
rozm*=2	$\lceil \log_2(rozm+n) \rceil$
end if	
return tab	1

Suma: $2 * (rozm+n) * \lceil \log_2(rozm+n) \rceil + 5n + 2 * \lceil \log_2(rozm+n) \rceil + 1$
dlatego algorytm działa w czasie $O(n \log n)$

2 Wyniki

Po przetestowaniu algorytmów otrzymano następujące wykresy:



3 Wnioski

Na wykresie oprócz przebiegów uzyskanych dzięki dokonanym pomiarom pokazano również przebiegi należące do klas $O(n^2)$ i $O(n \log n)$. Dzięki temu można zauważyć, że zgodnie z przeprowadzonymi wcześniej obliczeniami, algorytm powiększania tablicy o 1 ma asymptotyczne tempo wzrostu równe $O(n^2)$, natomiast algorytm podwajania rozmiaru tablicy - $O(n \log n)$. Warto zatem zwracać uwagę na wybór implementowanego algorytmu - oba działają na jednakowej strukturze, a ich złożoność obliczeniowa należy do różnych klas, co dla większej ilości danych powoduje znaczną różnicę w czasie wykonywania obliczeń (w przypadku przeprowadzonych testów różnica zaczyna być widoczna już dla ok. 10^5 wprowadzanych danych).