

Sprawozdanie laboratorium 6
Tablice asocjacyjne

1. Wstęp

Zadanie które mieliśmy wykonać na laboratoriach to struktura danych nazywana tablicą asocjacyjną bądź mapą, która przyjmuje obiekty do które posiadają swoją wartość oraz unikatowy klucz pozwalający się do nich dostać poprzez użycie ich jako indeksu w tablicy.

2. Realizacja

Struktura danych została stworzona jako klasa posiadająca tablice dynamiczną (upraszcza to tworzenie mapy o konkretnym rozmiarze) oraz funkcję hashującą. Każda wartość która ma zostać dodana do mapy zostaje „shashowana” a następnie umieszczona w tablicy. Do zbadania zostały użyte 2 funkcje hashujące, pierwsza elementarna polegająca na zliczeniu wartości liczbowych wszystkich znaków ANSII zadanego stringa oraz druga bardziej zaawansowana znaleziona w jednym z wielu repozytoriów na gicie. Pomiary zostały wykonane w zależności od ilości danych jakie przetrzymuje tablica, funkcji hashowania i rozmiaru samej mapy. Wartością wyszukiwania za każdym razem był ostatni wczytany element do mapy. Losowe elementy zostały wcześniej wygenerowane i zapisane do pliku „DATA”.

3. Wyniki

Hash 1

rozmiar tablicy	dane	ratio	czas[ns]
10	10	0,7	2206
100	100	0,5	2137
500	500	0,162	2023
1000	1000	0,089	3957
2000	2000	0,047	2189
3000	3000	0,031	4046
4000	4000	0,025	3118
5000	5000	0,02	2097
10000	10000	0,01	3603

rozmiar tablicy	dane	ratio	czas[ns]
11	10	0,63	3856
151	100	0,33	3864
541	500	0,14	3832
1423	1000	0,06	2010
2551	2000	0,03	4077
3361	3000	0,02	2235
4447	4000	0,022	2101
5563	5000	0,01	2538
10501	10000	0,01	2113

Tabela 1. Wyniki dla podstawowego hashowania

Hash 2

rozmiar tablicy	dane	ratio	czas[ns]
10	10	0,4	5037
100	100	0,24	1871
500	500	0,246	1805
1000	1000	0,125	2643
2000	2000	0,0625	2875
3000	3000	0,12	3607
4000	4000	0,062	2987
5000	5000	0,12	2091
10000	10000	0,06	3169

rozmiar tablicy	dane	ratio	czas[ns]
11	10	0,72	1857
151	100	0,49	1923
541	500	0,59	3368
1423	1000	0,5	2322
2551	2000	0,53	3523
3361	3000	0,59	1765
4447	4000	0,58	3467
5563	5000	0,59	2761
10501	10000	0,61	4033

Tabela 2. Pomiary dla Hashowania zaawansowanego

6. Funkcje hashujące

Funkcja podstawowa

```
int hash1(string s,int modulo){
    int h=0;
    if(s.length()>1){
        for(unsigned int i=0; i<s.length(); i++){
            h=h+((int)s[i]);
        }
    }else{
        h=(int)s[0];
    }
    return h % modulo;
}
```

Funkcja zaawansowana

```
unsigned int hash2(string s,int modulo){
    const char *tmp = s.c_str();
    const char* p;
    unsigned int g, h=0;
    for (p=tmp; (*p)!='\0'; p++){
        h= (h<<4) + (*p);
        if (g=h&0xF0000000){
            h=h^(g>>24); h=h^g;
        }
    }
    return h%modulo; }
```

Obie funkcje przyjmują jako argumenty ciąg znaków string oraz wartość moduła, która jest równa rozmiarowi mapy.

5. Wnioski

Na podstawie powyższych tabel można potwierdzić, że czas wyszukiwania wartości w tablicach asocjacyjnych utrzymuje stałe wartości z wahaniami nie przekraczającymi 2 mikro sekund.

Przy użyciu podstawowego hashowania zmiana rozmiaru tablicy nie przyczyniła się do zmiany współczynnika wypełnienia, który wraz ze wzrostem elementów liniowo spadał co może powodować wydłużenie czasu wyszukiwania oraz zbliżenie się jego notacji O do $O(n)$.

W przypadku zaawansowanej funkcji hashującej widać ogromny wpływ dobrania rozmiaru mapy. W pierwszym przypadku współczynnik wypełnienia spadał liniowo, zmiana rozmiarów mapy na liczbę pierwszą większą od ilości danych zaskutkowała bardzo dobrym współczynnikiem rozłożenia danych. Współczynnik utrzymywał się na poziomie 0,6 co jest bardzo dobrym wynikiem.