

Reikalavimai projekto Nr. 1 ataskaitai

1. Ataskaitoje turi būti pateikti algoritmų teoriniai įvertinimai jei jie egzistuoja literatūroje. Nurodyti iš kur paimti.
2. Suskaičiuotas algoritmo sudėtingumas, remiantis programos išeities tekstu.
Pvz.:

Atliekant įvertinimus su duomenų struktūromis ir objektais reikia atkreipti dėmesį, kad tiek klasių metodus tiek savybes reikia laikyti kaip procedūras ir jas įvertinti iš anksto. BubbleSort naudoja `MyDataArray` klasės `Length`, `operaciją[]` ir `Swap`.

	Kaina	Kartai
<pre>class MyDataArray { protected int length; public int Length { get { return length; } } double[] data; public MyDataArray(int n) { data = new double[n]; length = n; Random rand = new Random(); for (int i = 0; i < length; i++) { data[i] = rand.NextDouble(); } } public double this[int index] { get { return data[index]; } } public void Swap(int j, double a, double b) { data[j - 1] = a; data[j] = b; } }</pre>	c_1	1
	c_2	1
	c_3	1
	c_3	1

`MyDataArray` klasės atributo/savybės `Length` get įvertinimas $T_L(obj_MyDataArray) = c_1$, atributo pagal nutylėjimą $T_l(obj_MyDataArray, j) = c_2$, o metodo `Swap` sudėtingumas $T_S(obj_MyDataArray, j) = 2c_3$.

<code>static void BubbleSort(MydataArray items)</code>		
<code>{</code>		
<code>double prevdata, currentdata;</code>	c_4	1
<code>for (int i = items.Length - 1; i >= 0; i--)</code>	$T_L(items)$	1
<code>{</code>	c_5	items.Length + 1
<code>currentdata = items[0];</code>	$c_4 + T_L(items, 0)$	items.Length + 1
<code>for (int j = 1; j <= i; j++)</code>	c_5	$\sum_{i=0}^{items.Length} \sum_{j=1}^{i+1} 1$
<code>{</code>		
<code>prevdata = currentdata;</code>	c_7	$\sum_{i=0}^{items.Length} \sum_{j=1}^i 1$
<code>currentdata = items[j];</code>	$c_4 + T_L(items, j)$	$\sum_{i=0}^{items.Length} \sum_{j=1}^i 1$
<code>if (prevdata > currentdata)</code>	c_6	$\sum_{i=0}^{items.Length} \sum_{j=1}^i 1$
<code>{</code>		
<code>items.Swap(j, currentdata, prevdata);</code>	$c_4 + T_S(items, j)$	$\sum_{i=0}^{items.Length} \sum_{j=1}^i t_j$
<code>currentdata = prevdata;</code>	c_7	$\sum_{i=0}^{items.Length} \sum_{j=1}^i t_j$
<code>}</code>		
<code>}</code>		
<code>}</code>		

Raskime tarpines sumas:

$$\begin{aligned}
 \sum_{i=0}^{items.Length} \sum_{j=1}^{i+1} 1 &= \sum_{i=0}^{items.Length} (i+1) = \frac{items.Length(items.Length+1)}{2} + items.Length + 1 \\
 &= \frac{items.Length(items.Length+3)}{2} + 1 \\
 \sum_{i=0}^{items.Length} \sum_{j=1}^i 1 &= \sum_{i=0}^{items.Length} i = \frac{items.Length(items.Length+1)}{2} \\
 0 \leq \sum_{i=0}^{items.Length} \sum_{j=1}^i t_j &\leq \sum_{i=0}^{items.Length} \sum_{j=1}^i 1 = \frac{items.Length(items.Length+1)}{2}
 \end{aligned}$$

čia $t_j = 1$, jei sąlyga $prevdata > currentdata$ tenkinama, o kitu atveju $t_j = 0$.

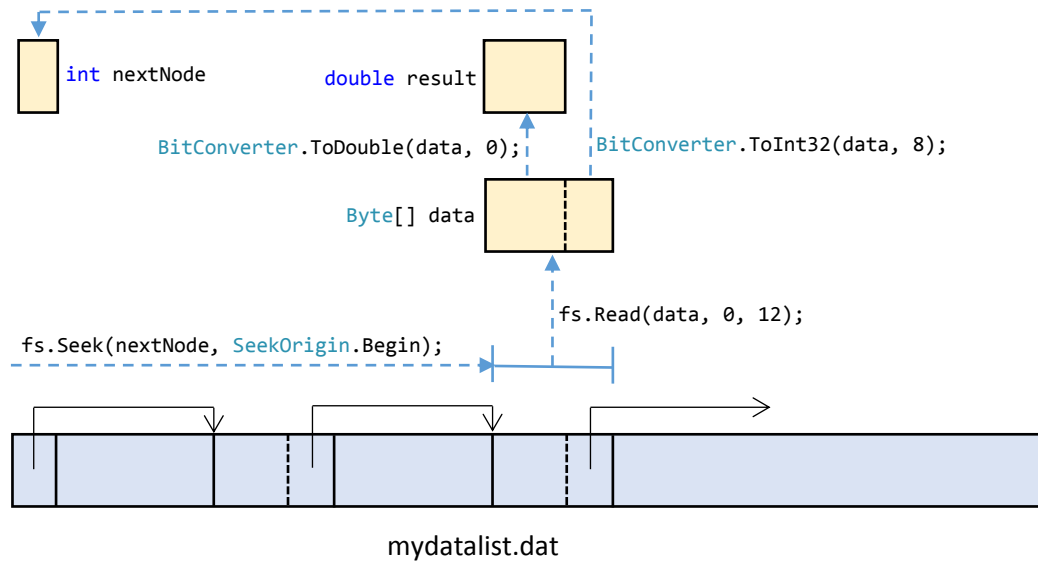
Tokiu atveju

$$\begin{aligned}
T_{Bubble}(obj_MyDataArray) &= c_4 + T_L(obj_MyDataArray) + c_5(items.Length + 1) \\
&+ (c_4 + T_I(items, 0))(items.Length + 1) + c_5 \frac{items.Length(items.Length + 3)}{2} \\
&+ c_7 \frac{items.Length(items.Length + 1)}{2} \\
&+ (c_4 + T_I(items, j)) \frac{items.Length(items.Length + 1)}{2} \\
&+ c_6 \frac{items.Length(items.Length + 1)}{2} + (c_4 + T_S(items, j)) \sum_{i=0}^{items.Length} \sum_{j=1}^i t_j \\
&+ c_7 \sum_{i=0}^{items.Length} \sum_{j=1}^i t_j \\
&= c_4 + c_1 + c_5(items.Length + 1) + (c_4 + c_2)(items.Length + 1) \\
&+ c_5 \frac{items.Length(items.Length + 3)}{2} + c_5 + c_7 \frac{items.Length(items.Length + 1)}{2} \\
&+ (c_4 + c_2) \frac{items.Length(items.Length + 1)}{2} + c_6 \frac{items.Length(items.Length + 1)}{2} \\
&+ (c_4 + 2c_3) \sum_{i=0}^{items.Length} \sum_{j=1}^i t_j + c_7 \sum_{i=0}^{items.Length} \sum_{j=1}^i t_j \\
&= \frac{c_2 + c_4 + c_5 + c_6 + c_7}{2} items.Length^2 \\
&+ \left(2c_2 + c_4 + 2c_5 + \frac{c_2 + c_4 + c_5 + c_6 + c_7}{2} \right) items.Length + (c_4 + 2c_3 + c_7) \sum_{i=0}^{items.Length} \sum_{j=1}^i t_j \\
&+ c_1 + c_2 + 2c_4 + c_5 = \Theta(items.Length^2)
\end{aligned}$$

nes nekeičia svariausio dėmes eilės sumos $\sum_{i=0}^{items.Length} \sum_{j=1}^i t_j$ įvertinimas tiek iš viršaus tiek iš apačios. Kadangi `MyDataArray` klasės konstruktorius turi parametą n , kuris nusako generuojamų duomenų kiekį, t. y. $Length = n$, tai $T_{Bubble}(obj_MyDataArray) = \Theta(n^2)$.

3. Pateikti viename iš pasirinktų algoritmų panaudotos duomenų struktūros realizacijos išorinėje atmintyje struktūrinę diagramą ir programos kodo fragmentą realizuojantį elemento nuskaitymą iš šios struktūros.

Pvz. Sąrašo duomenų struktūros realizavimo išorinėje atmintyje diagrama:



```

public override double Next()
{
    Byte[] data = new Byte[12];
    fs.Seek(nextNode, SeekOrigin.Begin);
    fs.Read(data, 0, 12);
    prevNode = currentNode;
    currentNode = nextNode;
    double result = BitConverter.ToDouble(data, 0);
    nextNode = BitConverter.ToInt32(data, 8);
    return result;
}
  
```

4. Privaloma aprašyti atliktus eksperimentus, o gautus rezultatus pavaizduoti lentelėmis bei grafikais.

>4. Analysis Bubble Sort

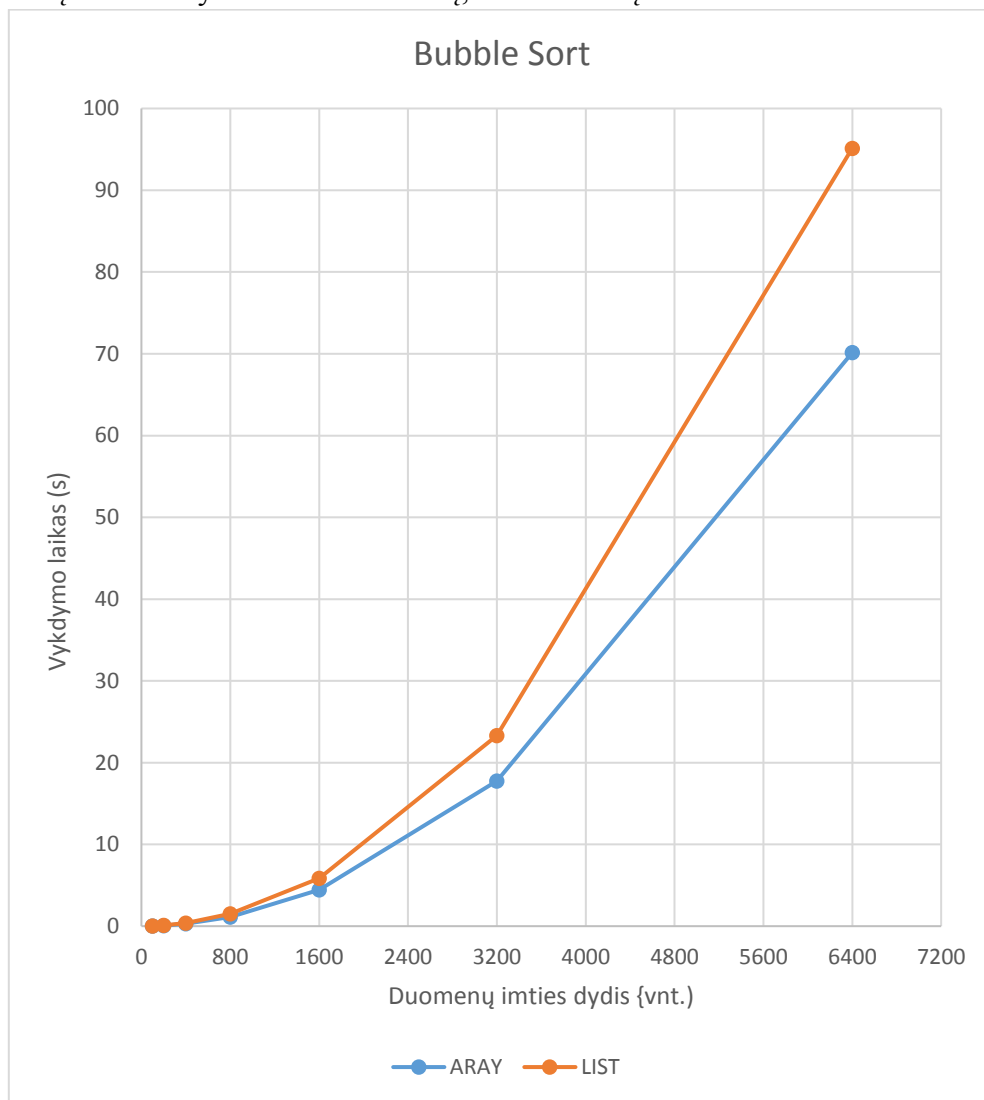
FILE ARRAY

N	Run Time	Op M Count	Op D Count
100	00:00:00.0193955	22 536	7 586
200	00:00:00.0705861	89 552	29 652
400	00:00:00.2964233	359 834	120 034
800	00:00:01.1320283	1 441 785	482 185
1 600	00:00:04.4384097	5 765 413	1 926 213
3 200	00:00:17.7471006	23 044 685	7 686 285
6 400	00:01:10.1451380	92 215 239	30 778 439

FILE LIST

N	Run Time	Op M Count	Op D Count
100	00:00:00.0248130	22 536	10 222
200	00:00:00.0896343	89 552	39 404
400	00:00:00.3733662	359 834	160 268
800	00:00:01.5027709	1 441 785	644 770
1 600	00:00:05.8692352	5 765 413	2 573 226
3 200	00:00:23.3155017	23 044 685	10 254 170
6 400	00:01:35.1427537	92 215 239	41 080 078

5. Braižant grafiką, horizontalioje ašyje atidedamos duomenų imtys, o vertikalioje – skaičiavimo laikai ar atliktų operacijų kiekis. Taškų padėtys turi atitikti skaitines koordinatų vertes (parinkit teisingą grafiko tipą „x y scatter“). Kiekvienas eksperimentas turi trukti nemažiau kelių sekundžių. Grafiką turi sudaryti nemažiau 7 taškų, didinant imtį vienodu koeficientu.



6. Ataskaitos gale turi būti pateikiamos prasmingos išvados, atitinkančios ataskaitos turinį.
7. Programų išeities tekstai, pateikiami priede.

Ataskaitos turinys

1. Užduotis.
2. Pasirinktos duomenų struktūros realizuotos išorinėje atmintyje struktūrinė diagrama.
3. Pirmojo rikiavimo algoritmo analizės rezultatai.
Teorinis bei suskaičiuotas algoritmo sudėtingumas, eksperimentiniai rezultatai pavaizduoti lentelėmis bei grafikais.
4. Antrojo rikiavimo algoritmo analizės rezultatai.
Teorinis bei suskaičiuotas algoritmo sudėtingumas, eksperimentiniai rezultatai pavaizduoti lentelėmis bei grafikais.
5. Paieškos algoritmo analizės rezultatai.
Teorinis bei suskaičiuotas algoritmo sudėtingumas, eksperimentiniai rezultatai pavaizduoti lentelėmis bei grafikais.
6. Išvados
7. Priedai
Programų išeities tekstai