Mikołaj Smolibowski

Bioinformatyka II rok

nr.145173

**Sprawozdanie z projektu 3 AKwB**

1. **Wczytanie sekwencji z pliku:**

Program otwiera plik o nazwie podawanej przez użytkownika. Pobierana jest nazwa sekwencji i tworzony jest nowy element klasy. W klasie przechowywany jest wektor przechowujący sekwencję, wektor przechowujący wartości poszczególnych nukleotydów, wektor przechowujący miejsce wystąpienia poszczególnych nukleotydów w sekwencji jak również zmienna do zapamiętania nazwy wczytanej sekwencji. Odczytana pierwsza linia pliku to nazwa sekwencji, która jest zapamiętywana dla elementu klasy. Następnie program zaczyna odczyt sekwencji znak po znaku. Odczytane nukleotydy są przekazywane do wektora przechowującego sekwencje. W momencie przekazania nukleotydu do wektora, przekazywana jest również jego pozycja w sekwencji poprzez podanie wartości inkrementatora do odpowiedniego wektora. Wartość inkrementatora jest zwiększana w momencie odczytu nowego nukleotydu. Gdy program natrafi na znak ‘>’ przerywa odczyt. Skonstruowany obiekt klasy przekazywany jest do wektora klasy, w którym przechowywane są obiekty. Następuje wczytanie nowej linii i utworzenie nowego obiektu klasy. Cały proces trwa do momentu natrafienia przez program znaku ‘;’.

W momencie jego wystąpienia, zamykany jest plik fast. Program przechodzi do odczytu pliku qual. W przypadku przygotowanych sekwencji ich wystąpienia w plikach fasta i qual są takie same. Dzięki temu przy odczycie 1 sekwencji z pliku qual wiadomo, że dana sekwencja w wektorze klasy ma miejsce vector[0]. Następuje pobranie pierwszej linii pliku tekstowego a następnie odczyt znak po znaku wartości nukleotydów. Wartości są odczytywane i przekazywane do wektora przechowującego wartości nukleotydów dla odpowiedniej sekwencji. Odczyt i przekazanie trwają tak długo, dopóki program nie natrafi na 0 które znajduje się na końcu każdej sekwencji jako sygnalizator do zaprzestania odczytu. W momencie wystąpienia 0 przerywana jest pętla wpisująca oraz zwiększana jest wartość inkrementatora, co pozwala na wczytanie kolejnych wartości do następnego obiektu klasy (następnej sekwencji). W momencie natrafienia w trakcie odczytu na symbol ‘;’ program zamyka plik oraz kończy odczyt.

1. **Modyfikacja sekwencji:**

Modyfikowanie sekwencji rozpoczyna się od podania przez użytkownika wartości wiarygodności. Wartość ta przekazywana jest do funkcji. Modyfikacja złożona jest z dwóch pętli for. Pierwsza pętla odpowiada za pobranie obiektu klasy. Druga natomiast za przeszukanie wektora wartości dla odpowiedniego obiektu. Jeśli wartość wiarygodności jakiegoś nukleotydu jest mniejsza od wartości podanej przez użytkownika program usuwa tą wartość z wektora. Oprócz usunięcia wartości wiarygodności nukleotydu z wektora następuje usunięcie nukleotydu z wektora sekwencji jak również usunięcie jego pozycji z wektora przechowującego te informacje. Po znalezieniu i usunięciu nukleotydu o wartości wiarygodności mniejszej niż ta podana pętla przeszukująca rozpoczyna się od nowa. Powtarzanie trwa tak długo aż w wektorze wartości zostaną tylko wystąpienia większe od wartości podanej przez użytkownika. Po zakończeniu pętli przeszukującej następuje przejście do następnej sekwencji i jej przeszukanie.

1. **Tworzenie wierzchołków**

Tworzenie wierzchołków rozpoczyna się od podania przez użytkownika  
wartości jakiej długości mają być tworzone podciągi. Wartość musi znajdować się w przedziale od 4 do 9. Wartość przekazywana jest do funkcji. Tworzenie wierzchołków złożone jest z 3 funkcji for. Pierwsza podobnie jak w przypadku modyfikacji odpowiada za wybranie obiektu klasy co pozwala na odczytania sekwencji jak również jej długości. W drugiej pętli tworzone są nowe obiekty klasy. Długość trwania pętli to wielkość sekwencji nukleotydowej pomniejszona o wartość długości podciągu podaną przez użytkownika. Do każdego obiektu przy wykorzystaniu trzeciej funkcji for wpisywana jest sekwencja o długości podanej przez użytkownika. Jak również do nowo powstałego obiektu przekazywana jest informacja, z której sekwencji pochodzi dany podciąg, numer pierwszego nukleotydu podciągu (numery podawane są po modyfikacji sekwencji oryginalnej), obiektowi nadawany jest również numer id oraz podana jest nazwa sekwencji oryginalnej, z której pochodzi podciąg. Tak utworzony wierzchołek (obiekt klasy) wpisywany jest do wektora wierzchołków. Następuje zwiększenie inkrementatora id i rozpoczynane jest tworzenie wierzchołka dla następnego nukleotydu sekwencji.

1. **Tworzenie połączeń w grafie**

Tworzenie połączeń pomiędzy wierzchołkami odbywa się poprzez porównanie ze sobą sekwencji podciągów wszystkich wierzchołków. Funkcja porównująca zbudowana jest z 3 funkcji for. Pierwsze dwie odpowiadają za pobranie porównywanych wierzchołków. W drugiej funkcji for dodatkowo znajduje się zmienna do zapamiętania ilości poprawnych wystąpień nukleotydów w sekwencji. Porównywanie wierzchołków odbywa się w momencie gdy nie są to te same wierzchołki (i != j). Jeśli są to dwa różne wierzchołki to następuje porównanie ich sekwencji przy użyciu trzeciej pętli for. W trzeciej pętli for znajduje się warunek sprawdzający czy wystąpienia nukleotydów porównywanych sekwencji są takie same. Jeśli nukleotydy w sekwencji są takie same zwiększana jest wartość zmiennej mającej pamiętać ilość poprawnych wystąpień nukleotydów. Po sprawdzeniu sekwencji obu wierzchołków sprawdzana jest wartość zmiennej opisanej wcześniej. Jeśli jej wartość jest równa długości sekwencji to oznacza, że porównywane sekwencje były takie same. Do wierzchołka pobranego w pierwszej pętli przekazywany jest numer wierzchołka, z którym był porównywany. Informacja ta przekazywana jest do wektora listy sąsiedztwa, który to wektor jest składową klasy wierzchołków. Następuje przejście do nowego wierzchołka, wyzerowanie zmiennej sprawdzającej poprawność wystąpień nukleotydów i sprawdzenie poprawności sekwencji kolejnego wierzchołka.

1. **Poszukiwanie kliki**

Dla poszukiwania kliki tworzony jest obiekt klasy. Do przechowania znalezionych klik użyty został wektor klasy *Cliq.* W klasie przechowywane są takie informacje jak:  
- sekwencja nukleotydowa znalezionej kliki,  
- elementy kliki (wierzchołki grafu),  
- numery podciągów składających się na klike,  
- nazwy sekwencji, z których pochodzą elementy kliki,  
- id kliki,

Funkcja poszukująca klikę bazuje na informacji, że w grafie łączone ze sobą wierzchołki o takiej samej sekwencji nukleotydowej podciągów i aby spełnić warunek znalezienia kliki, należy znaleźć taki zestaw aby w liście sąsiedztwa wystąpiły wierzchołki/podciągi pochodzące ze wszystkich wczytanych sekwencji sekwencji. W funkcji zadeklarowana jest tablica boolowska *sequences[5]*, której każdym miejscu przypisywana jest wartość. Jak również zmienna boolowska *fine* o wartości *true*. W pierwszej pętli wybierany jest wierzchołek dla, którego będzie sprawdzana lista następników. Jak również tworzony jest obiekt klasy. W następnej pętli sprawdzane jest, z której sekwencji pochodzi dany wierzchołek w liście, jak również sprawdzane jest z jakich sekwencji pochodzą wierzchołki obecne w liście następników sprawdzanego wierzchołka. W pętli tej umieszczone jest 5 warunków. Jeśli wystąpił wierzchołek z jednej z badanych sekwencji wtedy na odpowiednim miejscu tablicy *sequences* wartość *false* zamieniana jest na wartość *true*.

Po sprawdzeniu listy następników pętla for sprawdza, czy na którymś z miejsc tablicy *sequence[5]* nie występuje wartość *false*. Jeśli na, którymkolwiek miejscu w tablicy występuje wartość *false* wtedy zmieniana jest wartość zmiennej *fine* na *false*. Jeśli natomiast na wszystkich miejscach tablicy była wartość *true* wtedy spełniony jest warunek, mówiący że w tablicy następników występują elementy ze wszystkich sekwencji. Po spełnieniu tego warunku do obiektu klasy *cliquee* przekazywane jest:  
- sekwencja podciągu,  
- wpisywane są elementy kliki (wierzchołki grafu),  
- nadawane jest id kliki,  
Utworzony obiekt przekazywany jest do wektora klasy *cliquee*.

Po wykonaniu powyższych operacji do kliki przekazywane są informacje o numerze podciągu jak również nazwie sekwencji, z której pochodzi każdy podciąg należący do kliki. Całość wykonywana jest w 3 pętlach for gdzie dzięki znajomości id. wierzchołka należącego do kliki odnajdywany jest odpowiedni wierzchołek. Po jego znalezieniu do wektorów klasy *cliquee* przekazywana jest nazwa sekwencji jak również numer podciągu, które są zapisane w danym wierzchołku.

1. **Złożoność obliczeniowa**

Najbardziej złożonymi funkcjami w programie są:

- funkcja tworząca wierzchołki – 3 pętle for,

- funkcja tworząca połączenia w grafie – 3 pętle for,

- funkcja wyszukująca klike – 3 pętle for,

W samej funkcji wyszukującej klike umieszczone jest więcej pętli for, lecz nie są to pętle stricte zagnieżdżone w sobie lecz występujące po sobie. Dlatego też określiłbym je jako potrójne wystąpienie 2 pętli for.

Na tej podstawie szacuję złożoność wymienionych funkcji jako O(n3) każda.

Pozostałe funkcje:

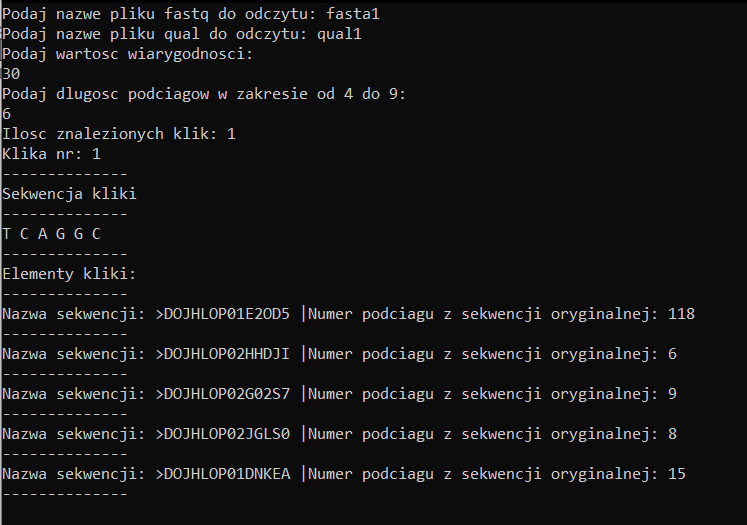
- funkcja modyfikująca sekwencje – 2 pętle for,

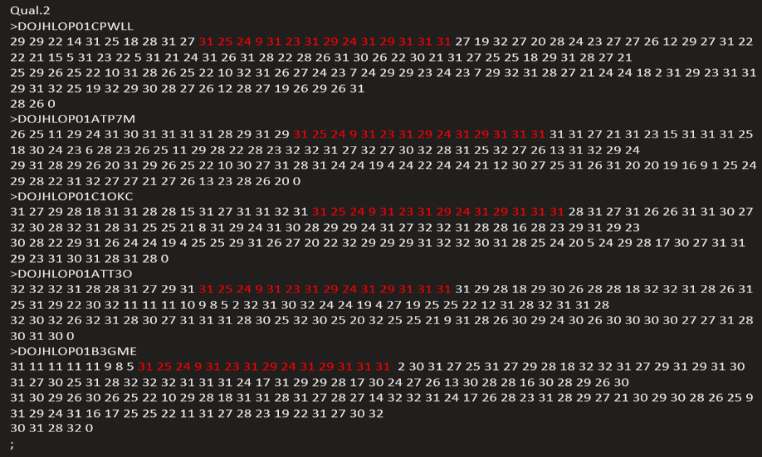
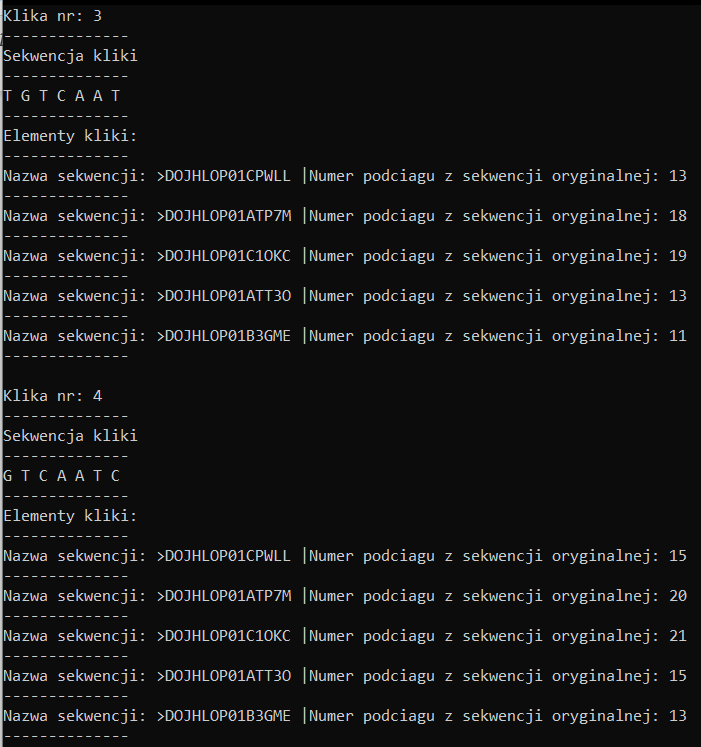
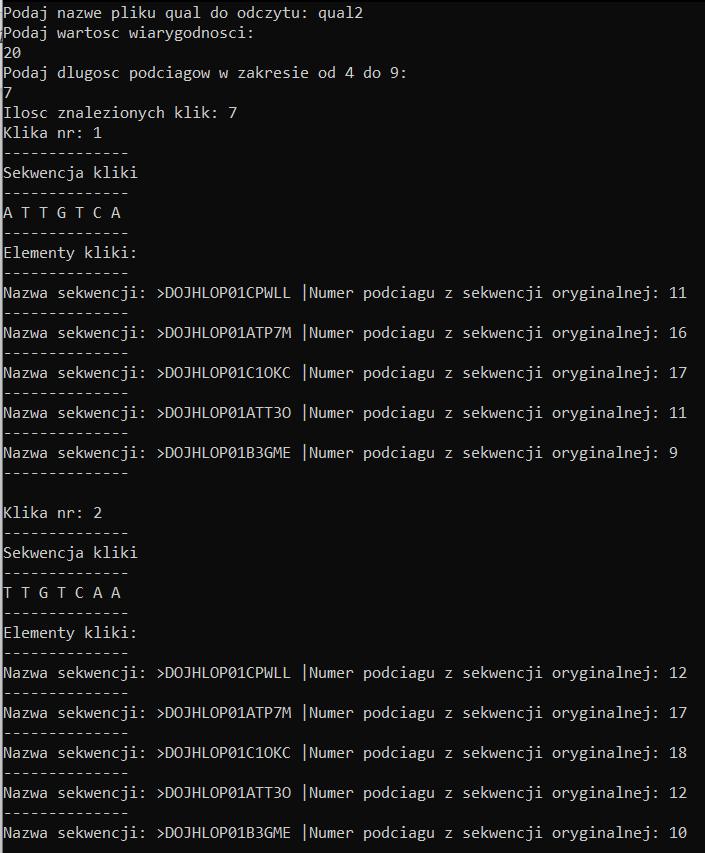
Złożoność obliczeniową całego programu szacuję jako: O(n3)

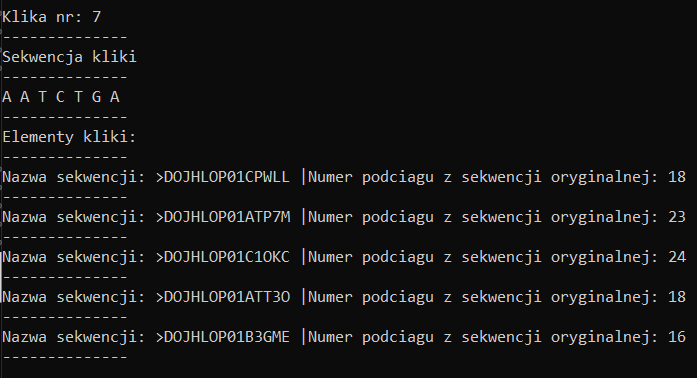
1. **Wyniki**

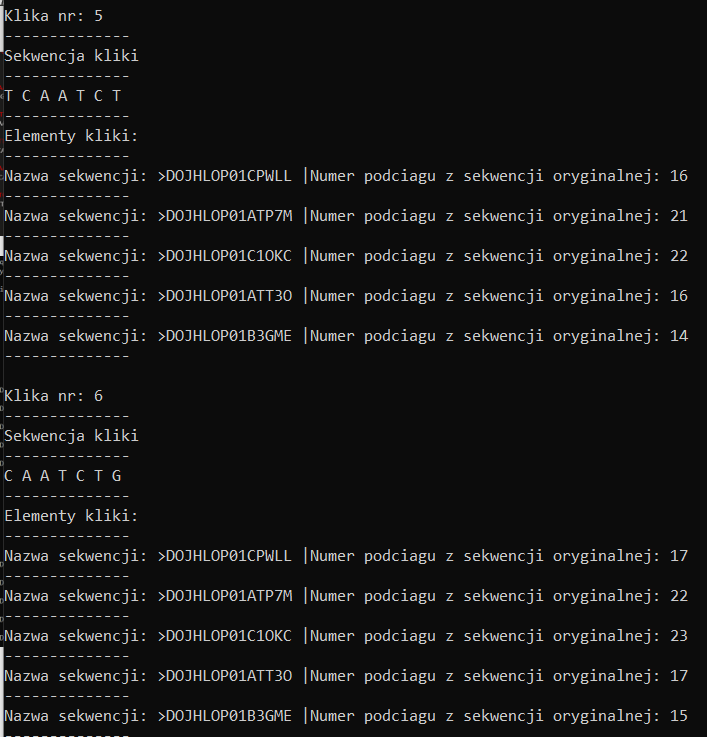


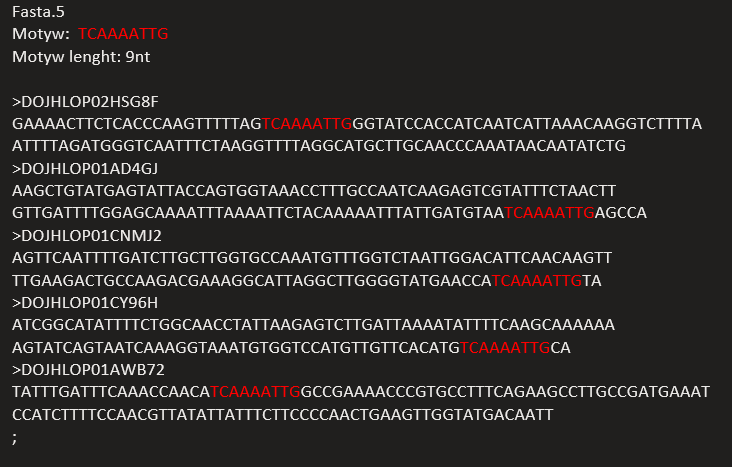


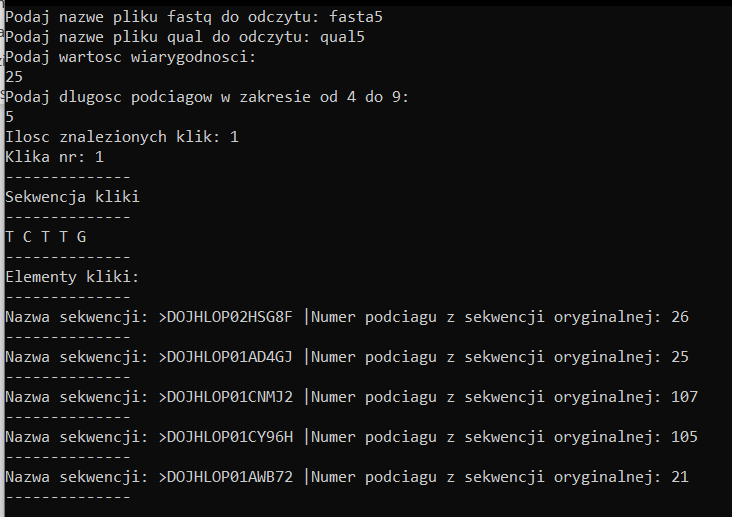












1. **Wnioski**

Można zauważyć, że znalezione kliki pokrywają się z motywami obecnymi w sekwencjach. Co było dla mnie ciekawe nie zawsze podanie mniejszej wiarygodności i krótszej sekwencji podciągów skutkowało znalezieniem większej ilości klik.

Program był bardzo ciekawy do napisania. Było to swego rodzaju namacalne wykorzystanie  
umiejętności pisania kodu do tworzenia programu do obróbki wyniku z sekwencjonowania (nawet jeśli miał to być tylko przykład i swego rodzaju wyzwanie-zabawa). Tak jak poprzednio największy problem sprawiło mi początkowe zrozumienie treści zadania. Lecz w momencie gdy tą treść na spokojnie zrozumiałem i przeanalizowałem to napisanie funkcji nie stanowiło już większego problemu. Ponieważ w większości opierało się to na odpowiednim znalezieniu obiektu/wektora/elementu w wektorze a następnie edycji czy przekazaniu wartości.