12주차 실습

2017-11-23

Topic

- Merge Sort
- Binary Search

프로젝트 - 포커게임

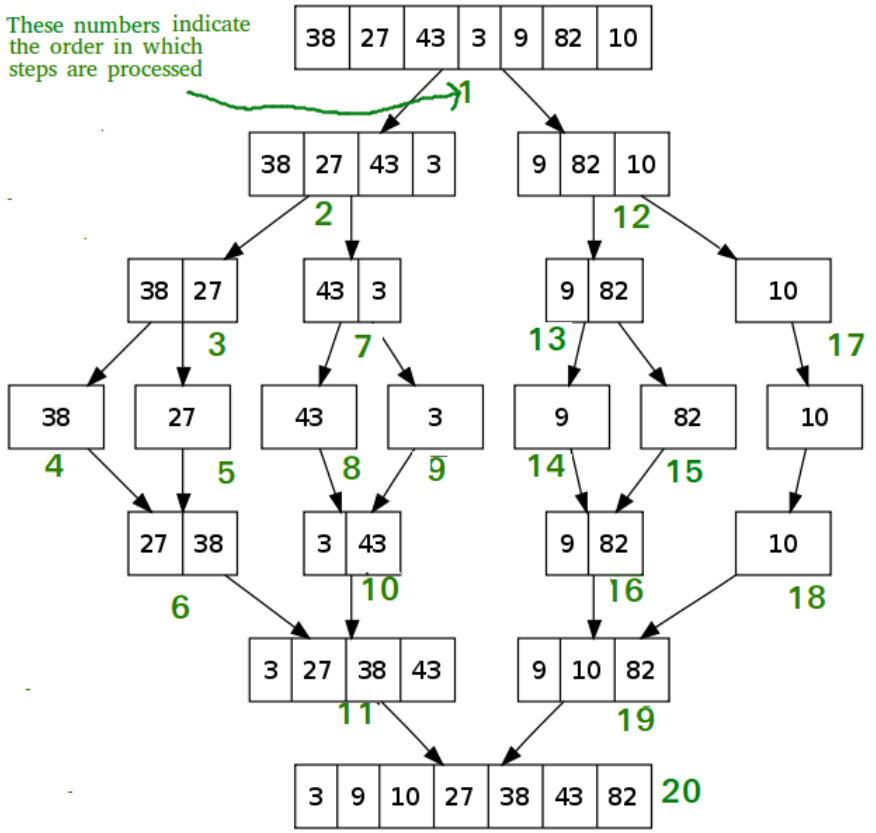
- 실습 4회에 걸쳐 부분적으로 진행할 예정입니다.
  - 11/23(목), 11/28(화), 11/30(목), 12/7(목)

- 이전 회차의 내용이 그 다음 회차에도 영향을 주기때문에, **그 다음 실습이 시작되기전에** 완성하여야 합니다.

Task 12_1 Merge Sort

Description

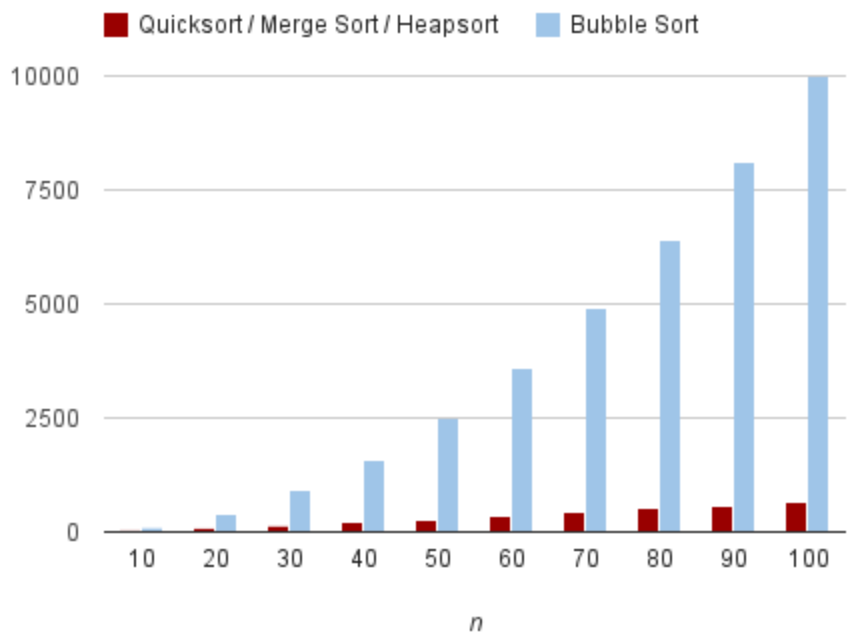
Merge sort is divide conquer algorithm, dividing an array into two arrays then sorting them each.



Task 12_1 Merge Sort

+ It works very fast than bubble or selection sort, as it's time complexity is O(nlogn), and others O(n^2)

In this problem, you'll implement merge sort.



Task 12_1 Merge Sort

Input

First line : number of elements(1 ~ 10,000,000)

Second line : integer array elements

Output

Array elements sorted by mergesort

input	output
5 3 2 4 1 5	1 2 3 4 5
5 0 0 0 0 0	0 0 0 0 0
5 -3 -2 -4 -1 -5	-5 -4 -3 -2 -1

Task 12_1 Merge Sort

Skeleton Code

```
#include <stdio.h>
#include <stdlib.h>

// Merges two subarrays of arr[].
// First subarray is arr[l..m]
// Second subarray is arr[m+1..r]
void merge( int arr[], int l, int m, int r ){
}
/* l is for left index and r is right index of the
sub-array of arr to be sorted */
void mergeSort( int arr[], int l, int r ){
}
```

```
// comment below when submit
int main(){
   int n;
   scanf( "%d", &n );
   int* arr = ( int *) malloc( sizeof( int ) * n);
   for( int i  =  0 ; i  <  n ; i++ ){
      scanf( "%d", &arr[i]);
   }
   mergeSort( arr, 0, n-1 );
   for( int i  =  0 ; i  <  n ; i++ ){
     printf( "%d ", arr[i] );
   }
   return 0;
}
```

Task 12_1 Merge Sort

You can easily download by

wget https://goo.gl/e7S3u2 -O fileName.c

Task 12_2 Cubic Equation

Description

Create a program to solve the cubic equation **using binary search**

ax^3 + bx^2 + cx + d = 0 (|x| < 10000, x : Integer)

Input

The first line gives the number of equations. (n < 10000)

From the second line, the coefficients of the equation (a, b, c, d)

Output

The Solutions of the equations are displayed on each line.

In a row, the solutions are **sorted in ascending order.**

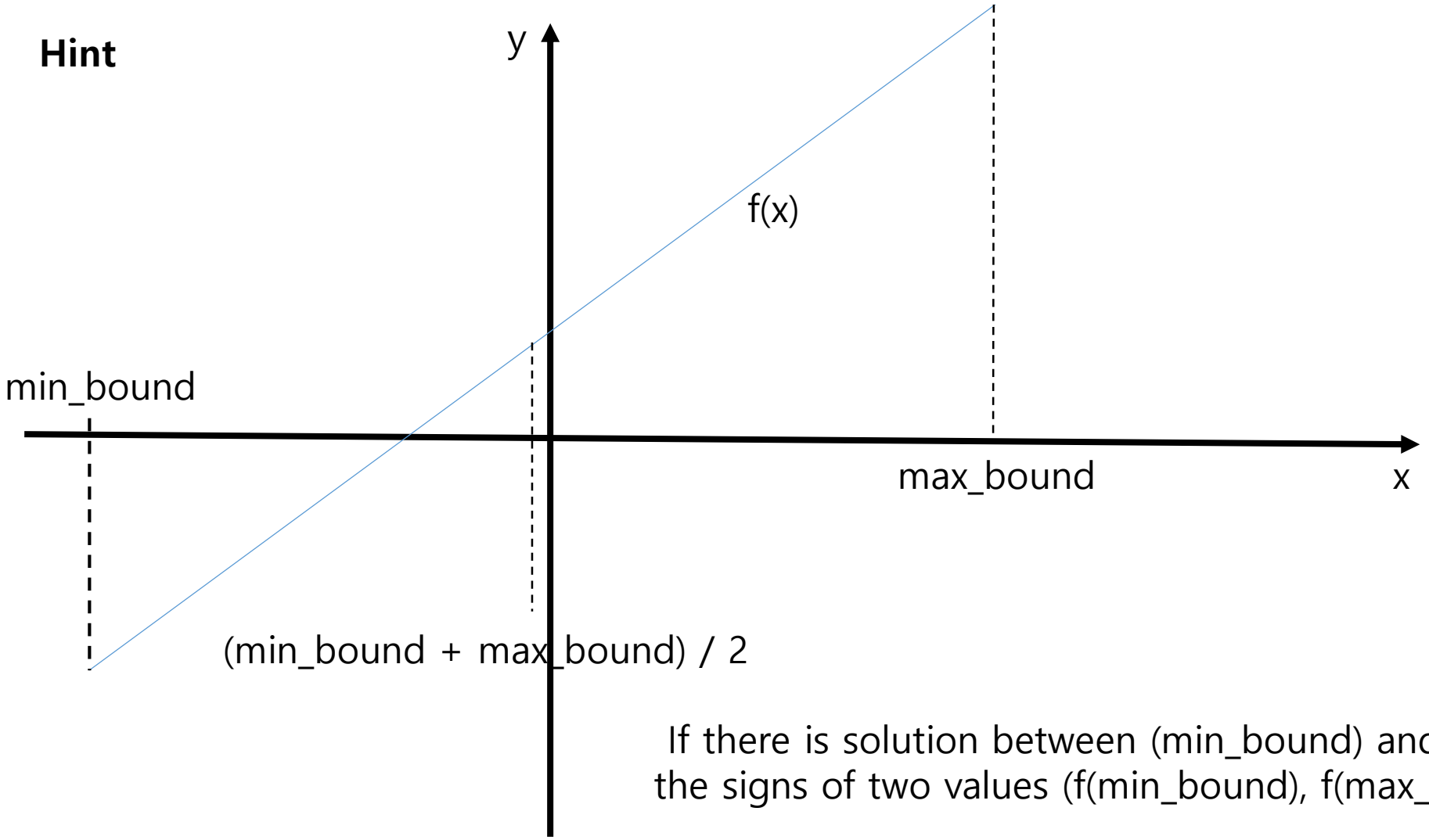
Task 12_2 Cubic Equation

Equation

Input	Output
2 100 0 0 0 1 -6 11 -6	0 1 2 3
2 1 -3 3 -1 1 -4 5 -2	1 1 2

Task 12_2 Cubic Equation

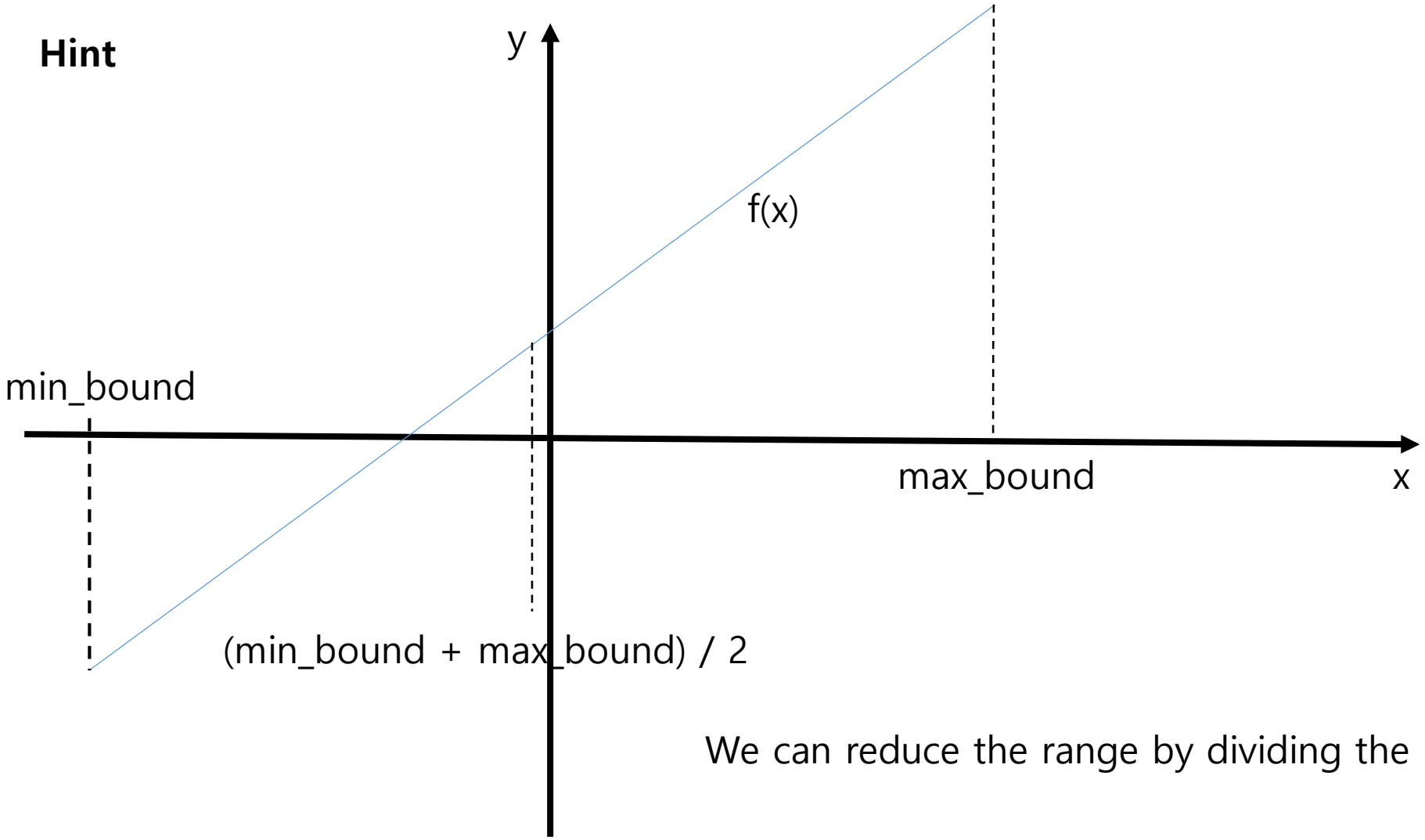
Hint



If there is solution between (min_bound) and (max_bound),
the signs of two values (f(min_bound), f(max_bound)) are different.

Task 12_2 Cubic Equation

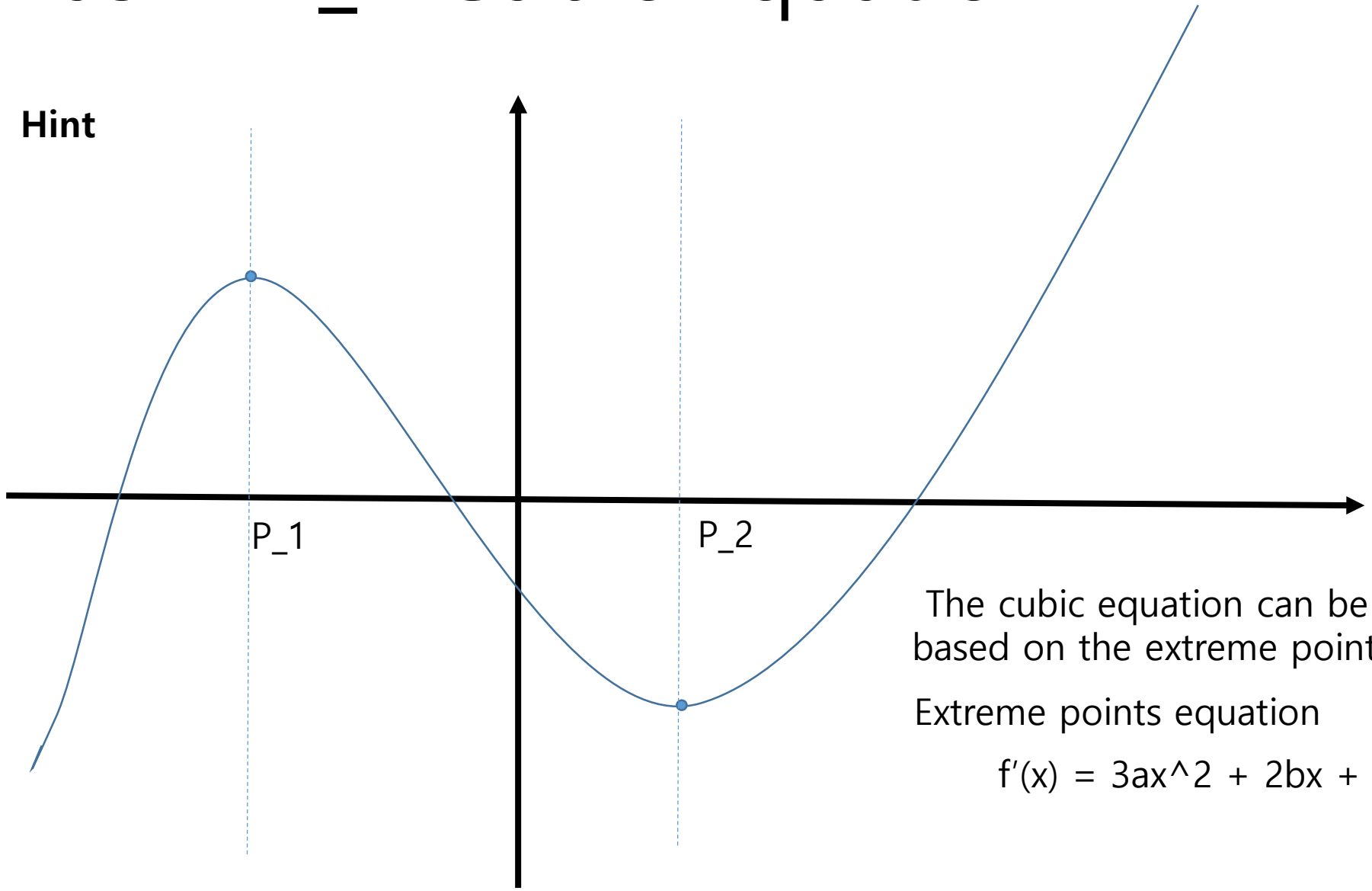
Hint



We can reduce the range by dividing the range by two.

Task 12_2 Cubic Equation

Hint



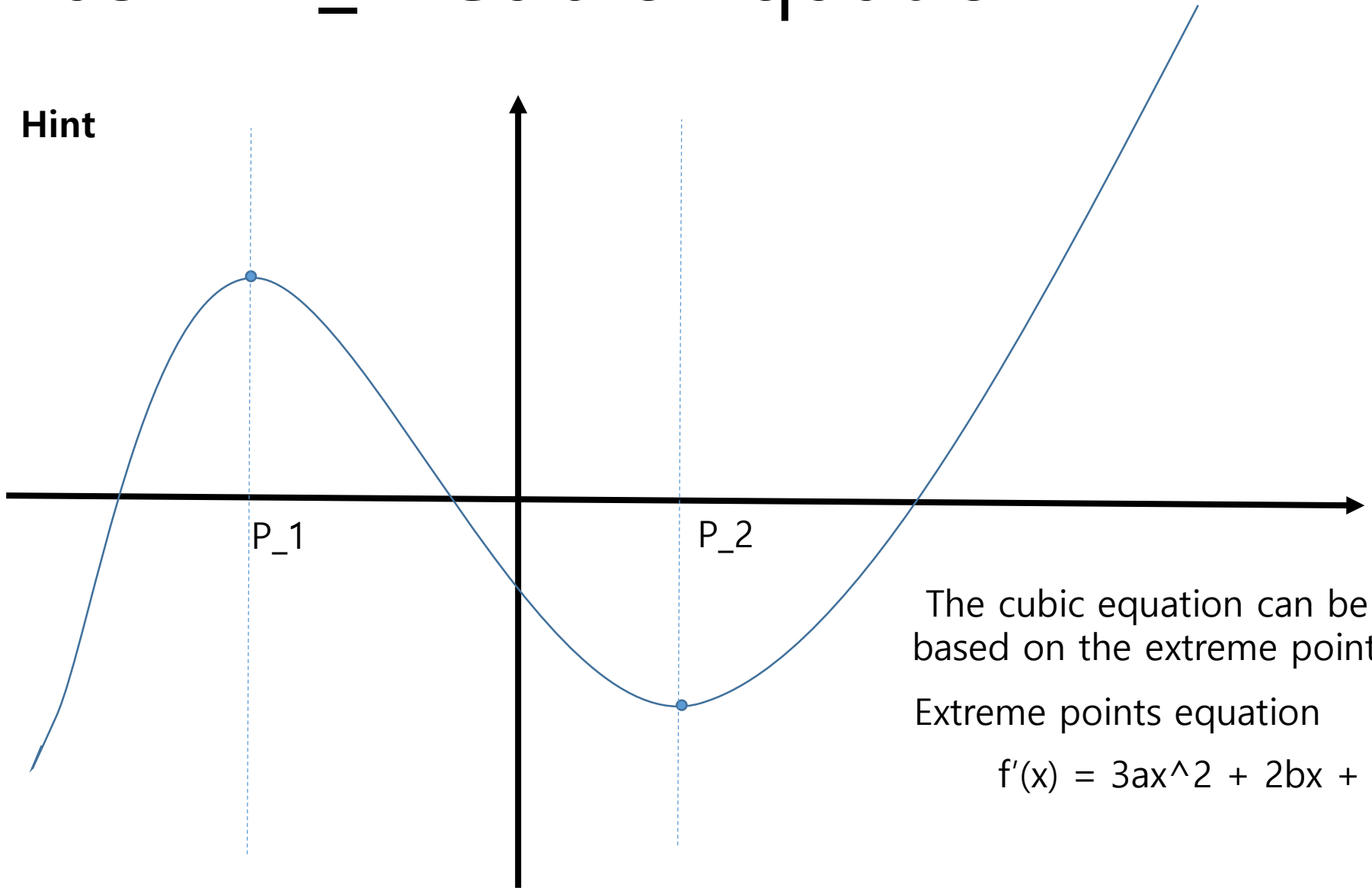
The cubic equation can be divided into sections based on the extreme points

Extreme points equation

$$f'(x) = 3ax^2 + 2bx + c = 0$$

Task 12_2 Cubic Equation

Hint



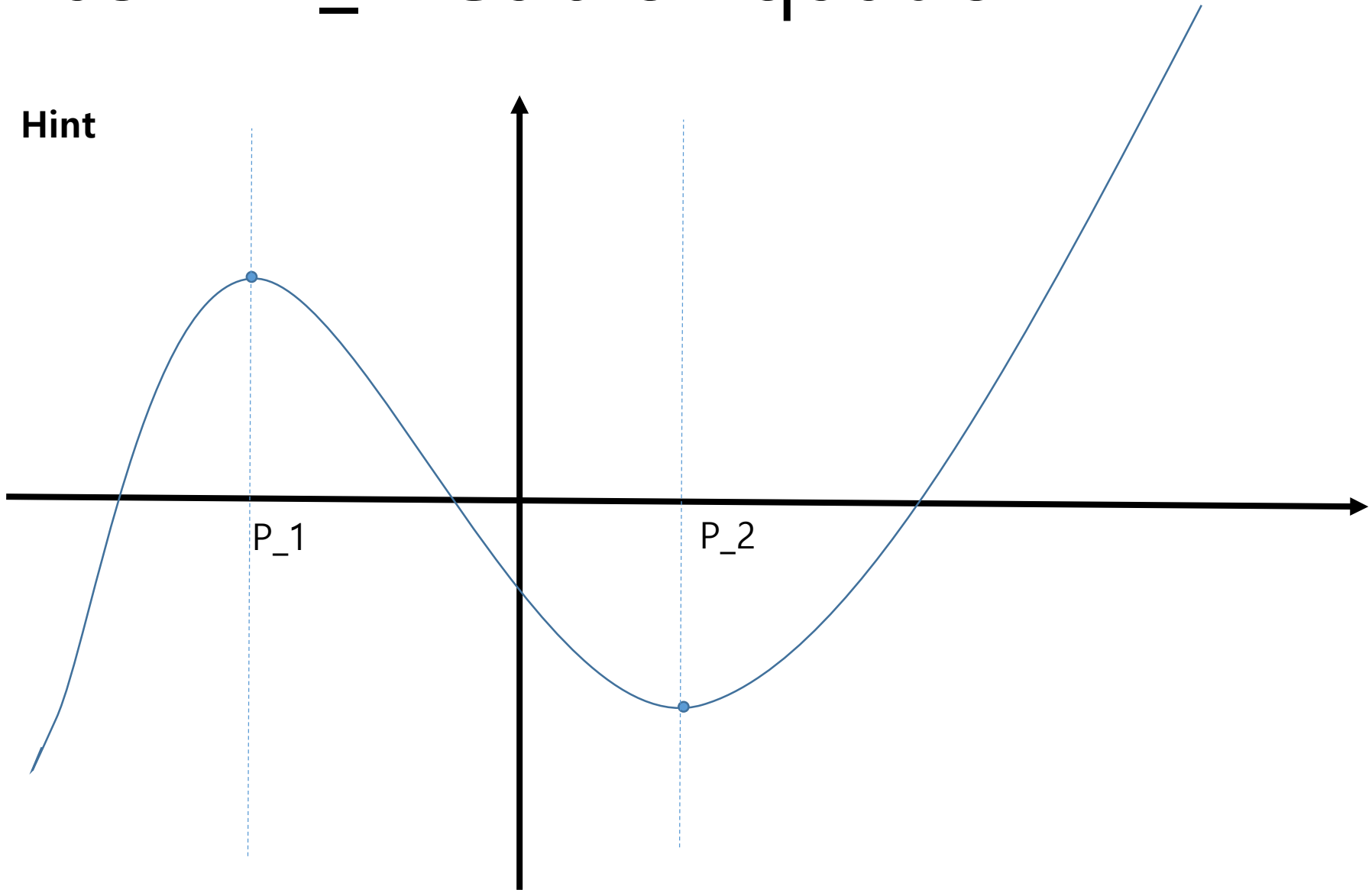
The cubic equation can be divided into sections based on the extreme points

Extreme points equation

$$f'(x) = 3ax^2 + 2bx + c = 0$$

Task 12_2 Cubic Equation

Hint



Poker Project

Poker Project

- We're going to make 1vs1 5-poker.
- Each player gets 5 cards, then start betting his/her money.
- After betting phase, players reveal their hands and player with higher rank wins (Takes money).
- We're going to implement poker AI later, and you can play with it!
- There are many rules in real world, but please follow rules described in this ppt. Otherwise, you might get no points.

Poker Project

- In this time, you will write a program that **determines rank of a player's hand**.
- In the second time, you will implement the **batting system** of Poker.
- In the third time, you will **combine the 1st, 2nd time implementations** and will make **interface of this game.**
- In the fourth time, you will implement **AI bot.**

Playing Cards

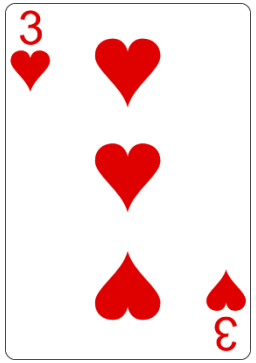
- 52 cards.
- We will use string to represent each cards.

- Symbol : Heart = ‘H’, Diamond = ‘D’, Club = ‘C’, Spade = ‘S’
- Number : Ace = ‘A’, Jack = ‘J’, Queen = ‘Q’, King = ‘K’

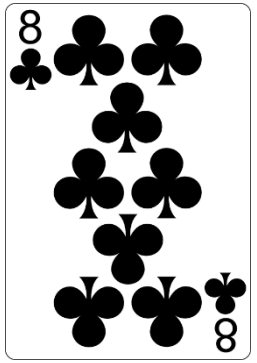
- See next page for better understanding.

Playing Cards (Examples)

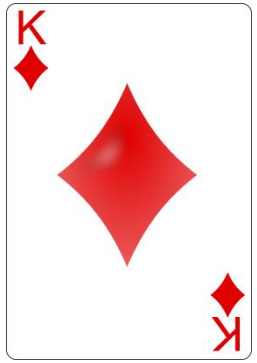
HK



H3



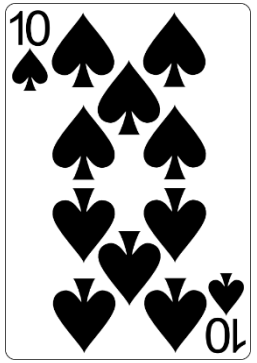
C8



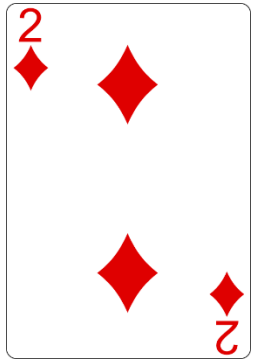
DK

HJ

SA

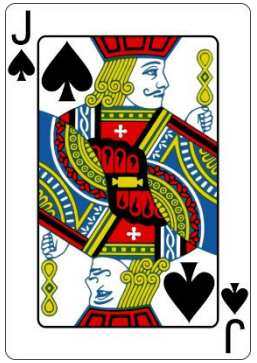


S10



D2

HQ



SJ

Determine Poker Hand

Higher rank wins

(rank 9) “Royal Straight Flush”
(rank 8) “Straight Flush”
(rank 7) “Four Card”
(rank 6) “Full House”
(rank 5) “Flush”
(rank 4) “Straight”
(rank 3) “Three Card”
(rank 2) “Two Pair”
(rank 1) “One Pair”
(rank 0) “No Pair”

You must use these strings.
Case-sensitive & blank-sensitive

Determine Poker Hand

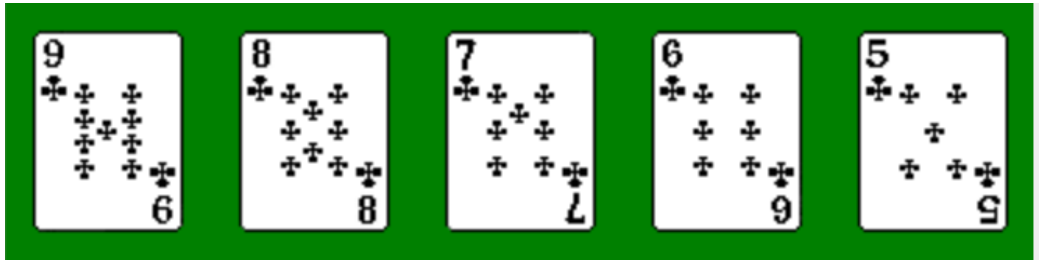
(rank 9) “Royal Straight Flush”

- ‘A,K,Q,J,10” with same symbols.

(rank 8) “Straight Flush”

- 5 consecutive numbers & same symbols.

* no back straight (A2345)



(rank 7) “Four Card”

- 4 same numbers.

Determine Poker Hand

(rank 6) “Full House”

- 3 same numbers + 2 same numbers.

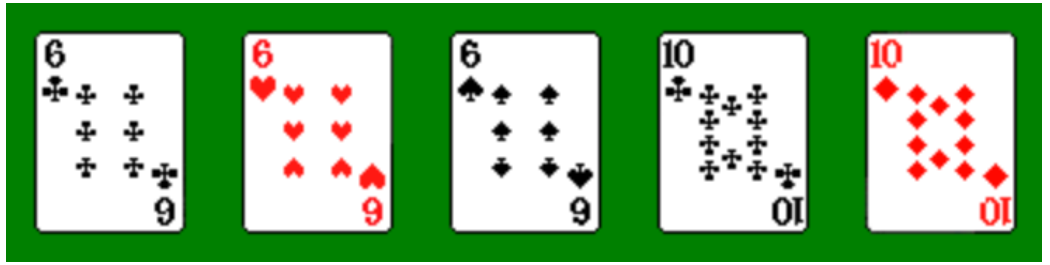
(rank 5) “Flush”

- 5 same symbols.

(rank 4) “Straight”

- 5 consecutive numbers.

* no back straight (A2345)



Determine Poker Hand

(rank 3) “Three Card”

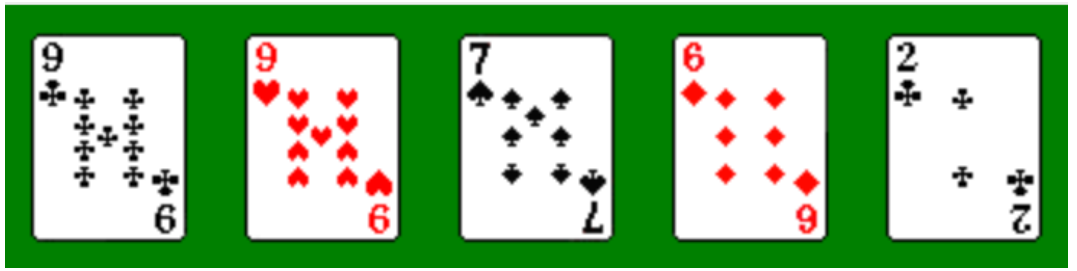
- 3 same numbers.

(rank 2) “Two Pair”

- 2 same numbers + 2 same numbers.

(rank 1) “One Pair”

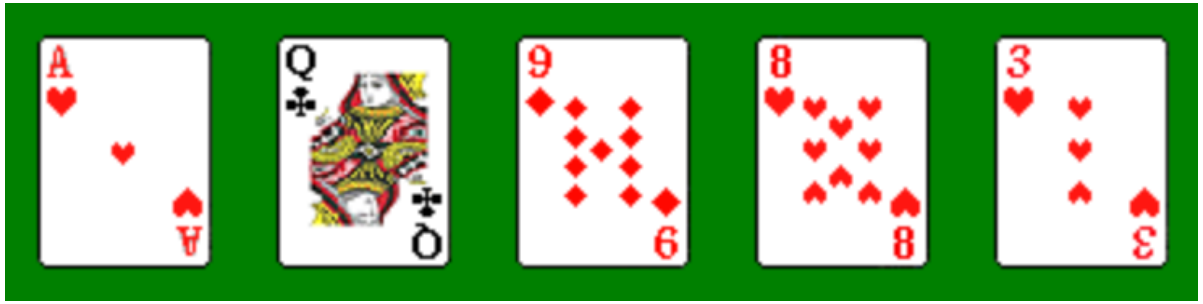
- 2 same numbers.



Determine Poker Hand

(rank 0) “No Pair”

- Nothing described above.



For more details, check

http://7poker.dreamx.com/game/7poker/7poker_help_01.asp

(But we do not consider back straight)

Hint:

You may consider J=11, Q=12, K=13, A=14

Task 12_3 Poker Project 1/4

Input

On the first line, number of test cases T are given. (0<= T <= 10000)

On the next T lines, 5 cards are given, separated by spaces.

Output

For each test cases, output type of hands. (Be aware of spellings)

See examples for detail.

Task 12_3 Poker Project 1/4

Input	**Output**
5 D4 H4 S4 C4 C5 D10 HA S3 C2 CA DA H10 SJ CQ CK DA DK DQ DJ D10 S2 S3 H2 H3 D10	Four Card One Pair Straight Royal Straight Flush Two Pair

Task 12_3 Poker Project 1/4

Input	Output
10 DA DK DQ DJ D10 C9 C8 C7 C6 C5 SK DK HK CK D10 C6 H6 S6 C10 D10 HA HJ H9 H7 H3 C10 H9 H8 S7 D6 C9 H9 D9 S7 D6 HK CK H7 S7 D6 C9 H9 S7 D6 C2 HA CQ D9 H8 H3	Royal Straight Flush Straight Flush Four Card Full House Flush Straight Three Card Two Pair One Pair No Pair