5주차 : Closest Pair of Points

알 고 리 즘

2015. 10. 15.

충남대학교 컴퓨터공학과 임베디드 시스템 연구실
TA 권진세

Overview

▶ Closest Pair of Points

– Closest Pair 란?

* 평면상의 n개 점이 있을 때 Euclidean Distance로 측정한 가장 근접한 점

– Euclidean Distance

$$\mathrm{d}(\mathbf{p},\mathbf{q}) = \mathrm{d}(\mathbf{q},\mathbf{p}) = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2 + \cdots + (q_n - p_n)^2}$$

$$= \sqrt{\sum_{i=1}^{n} (q_i - p_i)^2}.$$

* n은 차원

▶ 실습 / 과제

1) Divide and Conquer 를 이용한 **Closest Pair of Points** 구현

Closest Pair of Points

▶ Closest Pair of Points 란?

Divide-and-conque을 이용한 알고리즘.
분할, 정복, 결합 과정이 필요하다.

▷Divide

N개의 포인트를 ½로 나눈다.

▷Conquer

Left_min_dist : 좌측 편에 있는 포인트들 간의 최단 거리를 구한다
Right_min_dist : 우측 편에 있는 포인트들 간의 최단 거리를 구한다

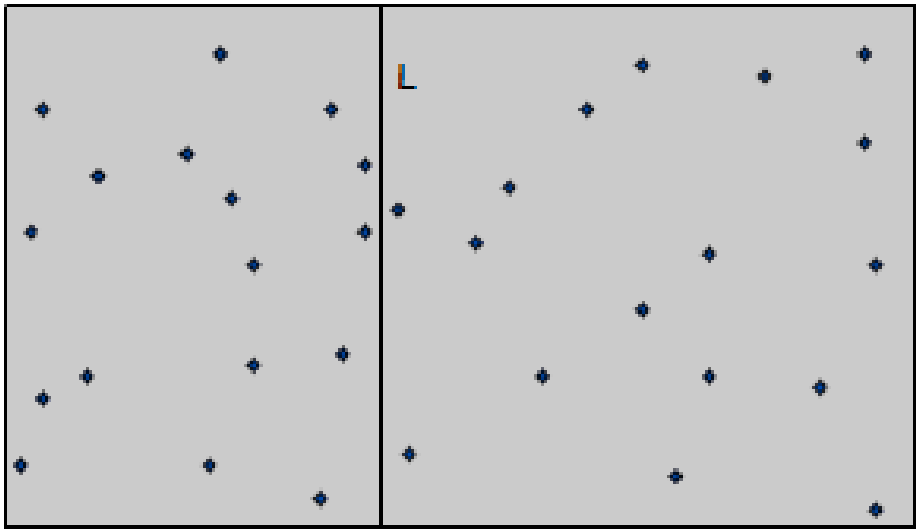
▷Combinde

중간점에 2δ 크기의 Window를 씌운다.
δ = min(Left_min_dist, Right_min_dist, inside_windw_dist)

Closest Pair of Points

▶ Closest Pair of Points 수행 과정

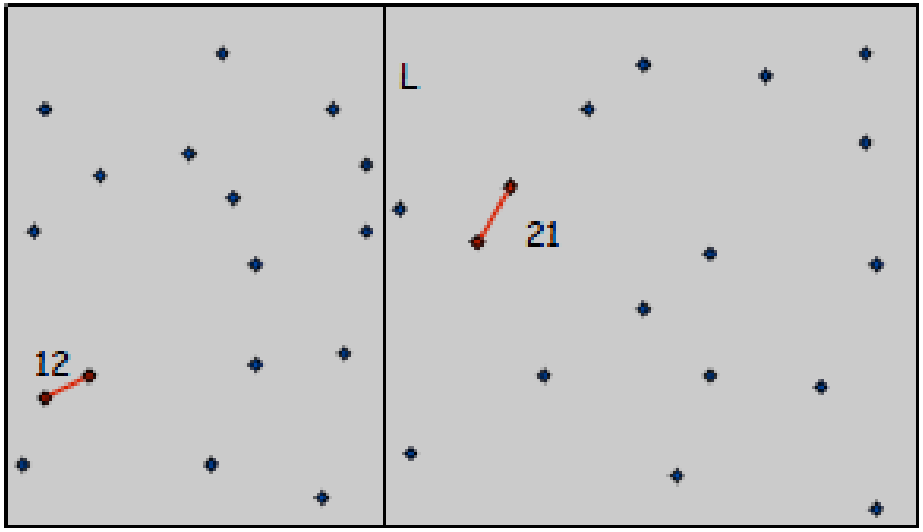
- ½ N 포인트로 중간 수직선을 그린다.



Closest Pair of Points

▶ Closest Pair of Points 수행 과정

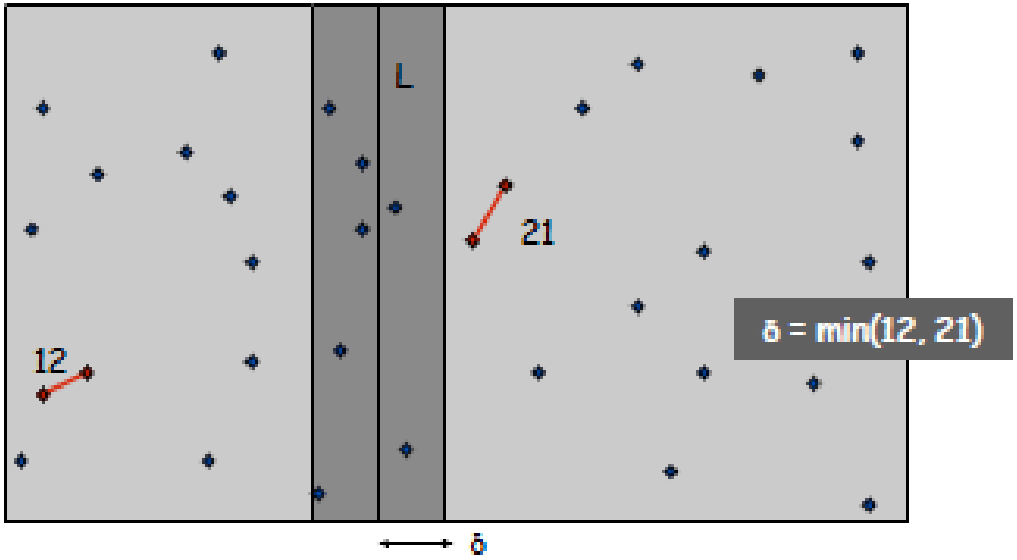
- 좌 / 우 측에서의 최단 거리를 각각 구한다.



Closest Pair of Points

▶ Closest Pair of Points 수행 과정

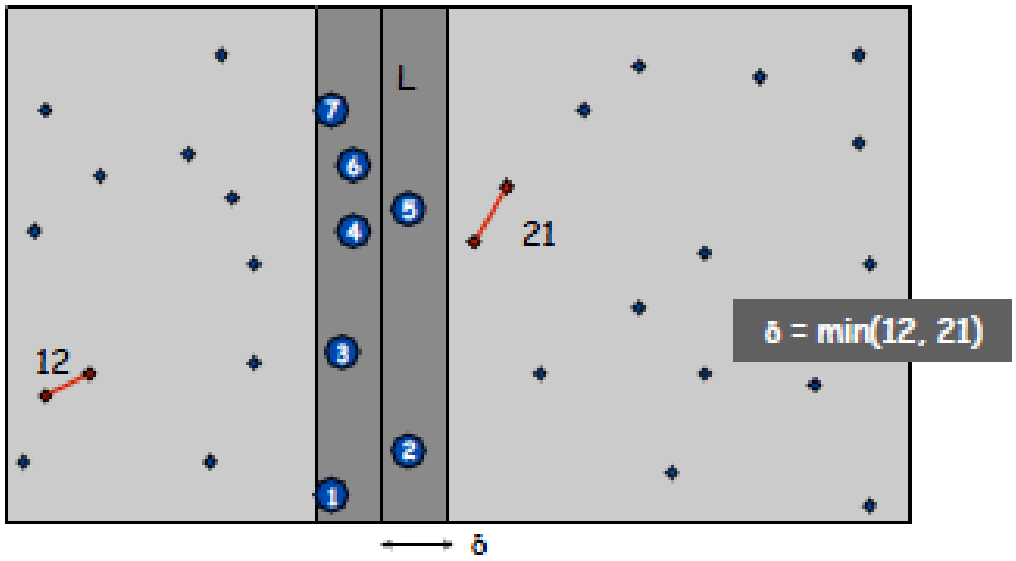
- 중간 점에 좌 / 우측 최단거리 중 최소값인 2δ Window를 씌운다



Closest Pair of Points

▶ Closest Pair of Points 수행 과정

- Window 내부에서 최단거리를 구한다



Closest Pair of Points

▶ Closest Pair of Points 수행 과정

- Window 내부에서 최단거리를 구하는 방법

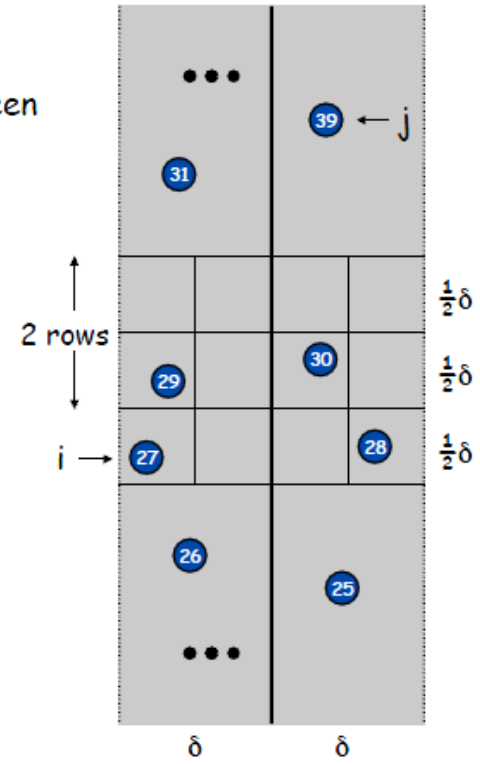
Def. Let s_i be the point in the 2δ-strip, with the i^{th} smallest y-coordinate.

Claim. If $|i - j| \geq 12$, then the distance between s_i and s_j is at least δ.

Pf.

- No two points lie in same $\frac{1}{2}\delta$-by-$\frac{1}{2}\delta$ box.
- Two points at least 2 rows apart have distance $\geq 2(\frac{1}{2}\delta)$. ▪

Fact. Still true if we replace 12 with 7.



Closest Pair of Points

▶ Closest Pair of Points 알고리즘

```
Closest-Pair(p1, …, pn) {
   Compute separation line L such that half the points
   are on one side and half on the other side.

   δ1 = Closest-Pair(left half)
   δ2 = Closest-Pair(right half)
   δ  = min(δ1, δ2)

   Delete all points further than δ from separation line L

   Sort remaining points by y-coordinate.

   Scan points in y-order and compare distance between
   each point
   If any of these distances is less than δ, update δ.

   return δ.
}
```

$O(n \log n)$

$2T(n / 2)$

$O(n)$

$O(n \log n)$

$O(n)$

Closest Pair of Points

▶ Closest Pair of Points 실제 구현

```
Closest-Pair(p1, …, pn) {
   Compute separation line L such that half the points
   are on one side and half on the other side.

   δ1 = Closest-Pair(left half)
   δ2 = Closest-Pair(right half)
   δ  = min(δ1, δ2)

   Delete all points further than δ from separation line L

   Sort remaining points by y-coordinate.

   Scan points in y-order and compare distance between
   each point
   If any of these distances is less than δ, update δ.

   return δ.
}
```

2차원 평면상의 배열을 x에 대하여 sort를 수행한다.

삭제 하기 보다는 새로운 배열을 만들어 값을 넣어준다.

Y값을 기준으로 Sorting

Window 내부의 최단거리를 구한다.

(Y값을 기준으로 δ 값보다 작은 거리에 있는 값들만 비교한다.)

Brute force. Check all pairs of points p and q with $\Theta(n^2)$ comparisons.

Loop invariant로 termination condition시
모든 포인트에 대하여 반복하여 거리를 구한다.

Practice / Homework

1. Closest Pair of Points 구현

과제 예시)

Input Data : 1.23 12.3, 1.0 2.0, 3.1 21.2,5.2 10.0
Output Data : 4.588

Practice / Homework

※ 그 외 실습 과제 수행 중 유의 사항

- 포함내용 : 코드와 보고서만 제출
  ※ 매주 실습 보고서도 함께 제출할 것
  ※ 201500000_05.c 파일 함께 압축 후 전송

- 제출이름 :
  메일 : [알고리즘00반]_201500000_홍길동_5주차
  파일 : 201500000_05.c / 보고서_201500000_홍길동_05주차.hwp

- 제출기한 : 2015-10-22 18:00까지

- 메일주소 : kwonse@cnu.ac.kr

APPENDIX 1. File I/O

1. 파일 입출력 방법

```
FILE* fp;          //fp : input file pointer
FILE* fop;         //fop : input file pointer

//파일 이름은 "00_201500000_insertion.txt"
//입출력 파일은 *.c 소스파일과 같은 폴더에 있어야 한다.
fp = fopen(FILENAME,"rt");                  //입력 파일 열기
fop = fopen(FILENAME2,"wt");                //출력 파일 열기

if (fp == NULL) {
        printf("**** Input File open error ****₩n");
        exit(1);
} //파일 없을 경우 예외처리로 프로그램 종료
```

APPENDIX 1. File I/O (계속)

```
while(!feof(fp)){
} //파일의 끝날때 까지 반복

fscanf(fp,"%d, ", &변수);        // ex) 123, 456,
                                 // int 값만 추출함 ', '는 제외됨

fprintf(fop, "%d", 출력할 값);  // 파일 출력 시 사용

fclose(fp);
```

APPENDIX 2. 배열 넘기기

1. Main 함수의 배열을 주소로 넘겨서 다룰 때 !! 이중포인터 사용
   - 장점 : 메모리 절약, 리턴 불필요.

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

void user_malloc(int** num);

void main(void){
    int *ptr;
    user_malloc(&ptr);      //포인터 변수 ptr의 주소를 인자로 보냄.
    printf("%d₩n", *ptr);  //출력 값은 10 이다.
    return 0;
}

void user_malloc(int** num){
    *num = (int*)malloc(sizeof(int));
    (*num)[0] = 10;
}
```

APPENDIX 3. 동적 할당 메모리 크기

Q & A.

포인터로 받은 배열의 크기를 구하는 방법? (있다)

- Malloc 함수의 선언을 보면 void* malloc(size_t size)
- Size_t는 많은경우 unsinged long int로 되어있으므로
- 메모리를 할당 할 때 이 크기만큼 더 할당해서 할당 영역의 처음 부분에 길이의 값을 저장해 두고 있음.

- *(ptr - sizeof(size_t))
- **다음과 같은 함수로 만들어 사용 가능**

```
int sizeof_ar(int* S){
    int size;
    size = *(S - sizeof(int));
    return size;
}
```

APPENDIX 4. 중간 값 찾기

3개의 원소중에 중간 값을 찾는 방법

```
int iPivot;
int ptrCenter = (ptrLeft + ptrRight) / 2;
if(!(ptrLeft < ptrCenter ^ ptrCenter < ptrRight))
    iPivot = ptrCenter;
else if(!(ptrCenter < ptrLeft ^ ptrLeft < ptrRight))
    iPivot = ptrLeft;
else
    iPivot = ptrRight;
```

APPENDIX 5. quick sort lib func

Quick sort library function

#include <stdlib.h>

```
int compareX(const void* a, const void* b)
{
    d2_arr *p1 = (d2_arr *)a,  *p2 = (d2_arr *)b;
    return (p1->x - p2->x);
}
int compareY(const void* a, const void* b)
{
    d2_arr *p1 = (d2_arr *)a,  *p2 = (d2_arr *)b;
    return (p1->y - p2->y);
}
```

qsort(arr, arr size, element size, compare_위에참조)

APPENDIX 6. 각 자료형의 최대크기

Variant limits 헤더

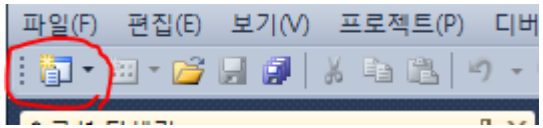
#include <limits.h>
=>정수형 변수의 최대값을 전처리 매크로로 저장한 헤더

#include <float.h>
⇒ ex) double 사이즈의 최대 크기를 알고 싶을 때
⇒ printf("%lf",DBL_MAX);

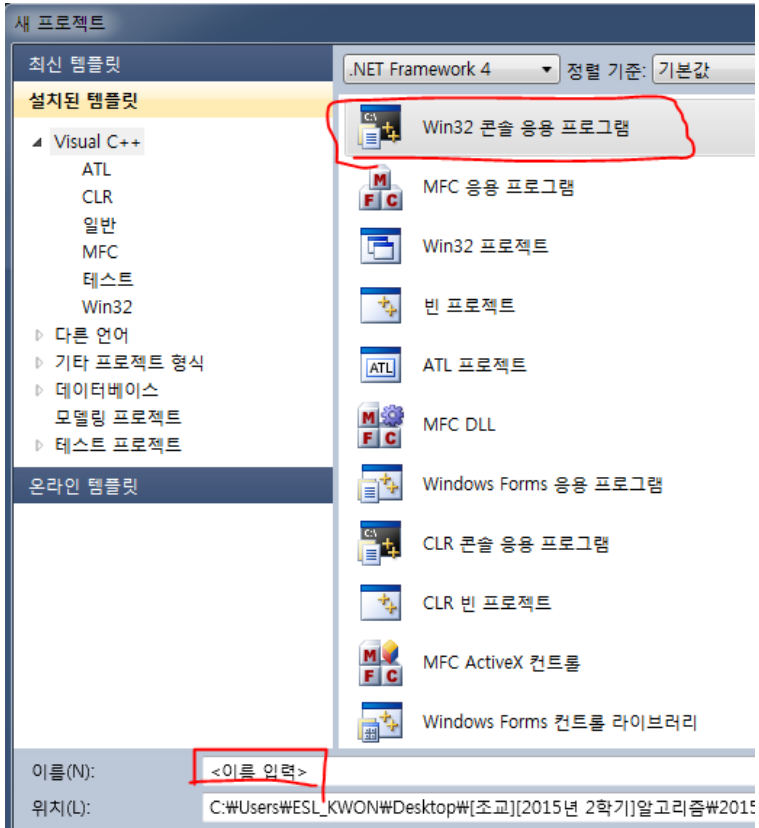
Trouble Shooting

Visual Studio 2010 프로젝트 생성

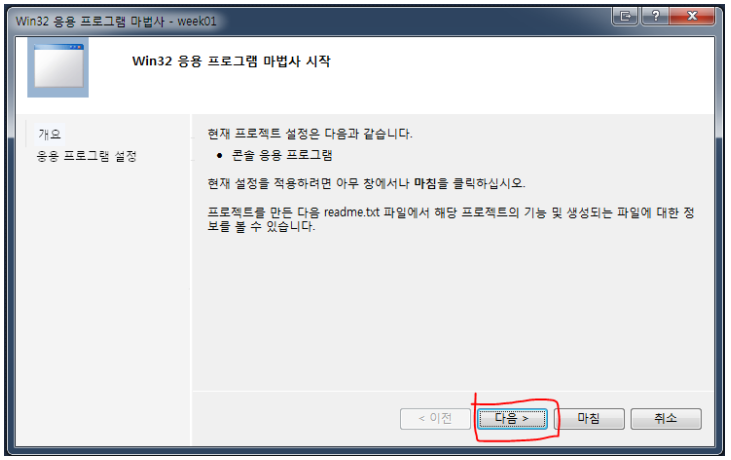
1.



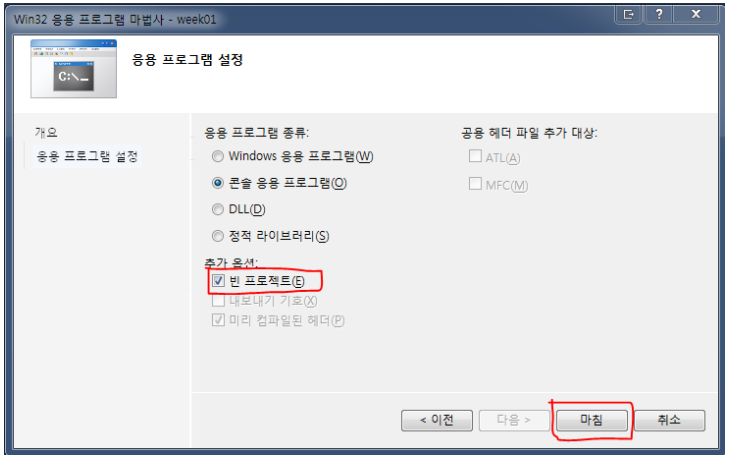
2.



3.



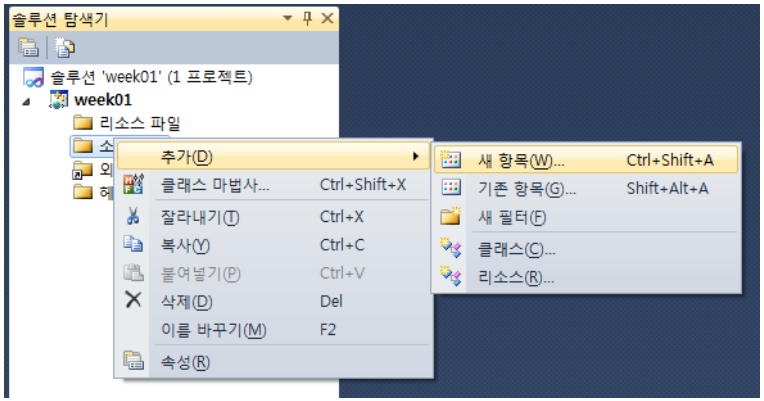
4.



Trouble Shooting

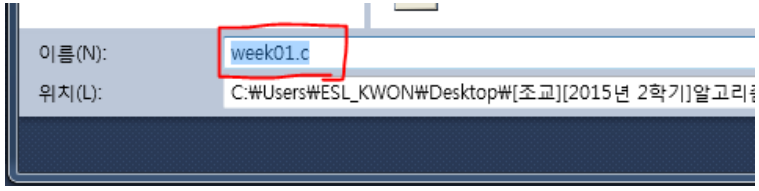
Visual Studio 2010 프로젝트 생성

5.



6.

7.



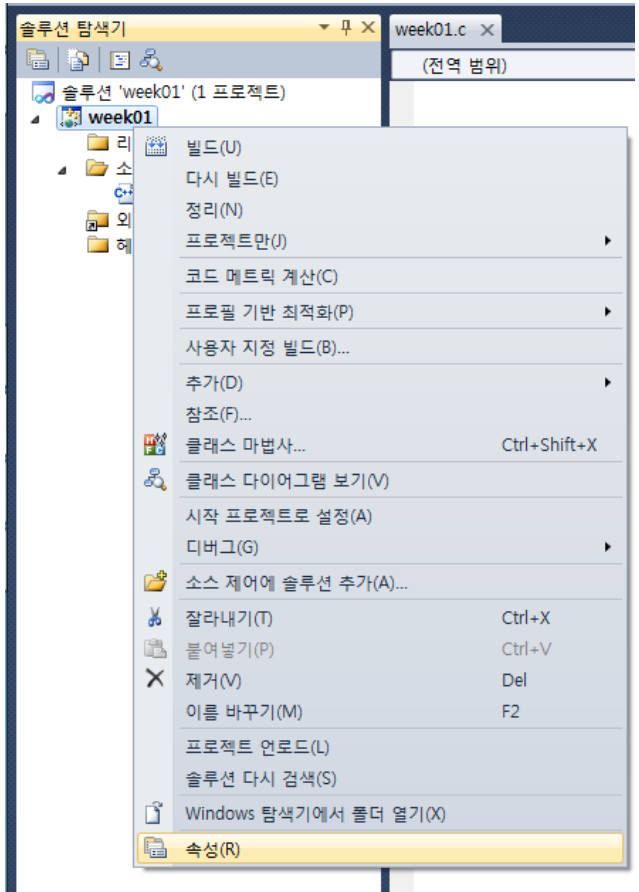
반드시 .c 로 이름 변경!!

Trouble Shooting

Visual Studio 2010 메니페스트 오류 해결

```
1>LINK : fatal error LNK1123: COFF로 변환하는 동안 오류가 발생했습니다. 파일이 잘못되었거나 손상되었습니다.
1>
1>빌드하지 못했습니다.
```

1.

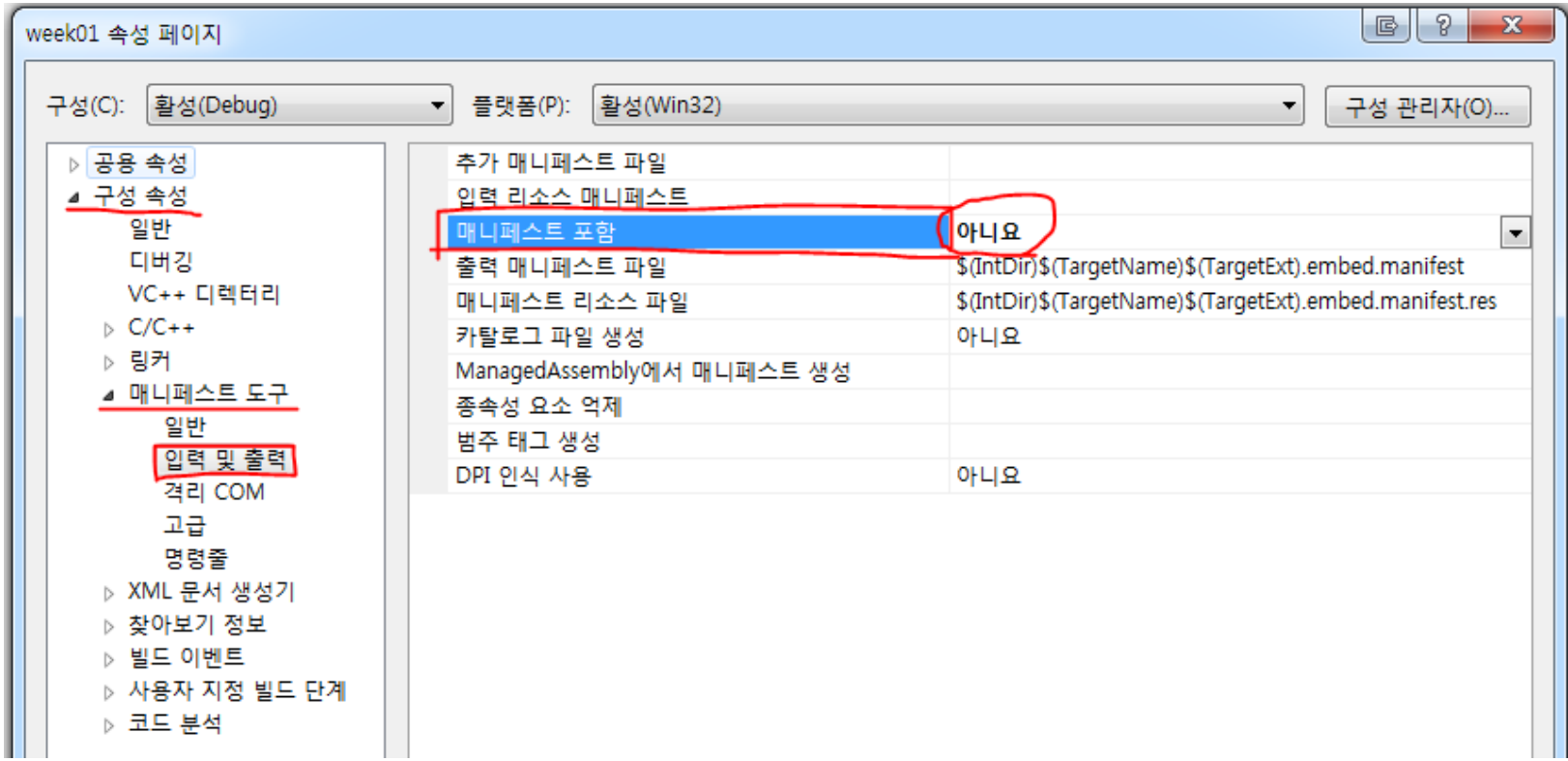


< 프로젝트 속성열기

Trouble Shooting

Visual Studio 2010 메니페스트 오류 해결

2.



구성 속성 -> 매니페스트 도구 -> 입력 및 출력 ->

메니페스트포함 : “아니요 "

Trouble Shooting

Visual Studio 2010 메니페스트 오류 해결
3. 매니페스트 문제 영구적 해결 방법

Visual Studio Service Pack 1 다운로드.
(>600MB 오래 걸림…)

https://www.microsoft.com/en-us/download/confirmation.aspx?id=23691