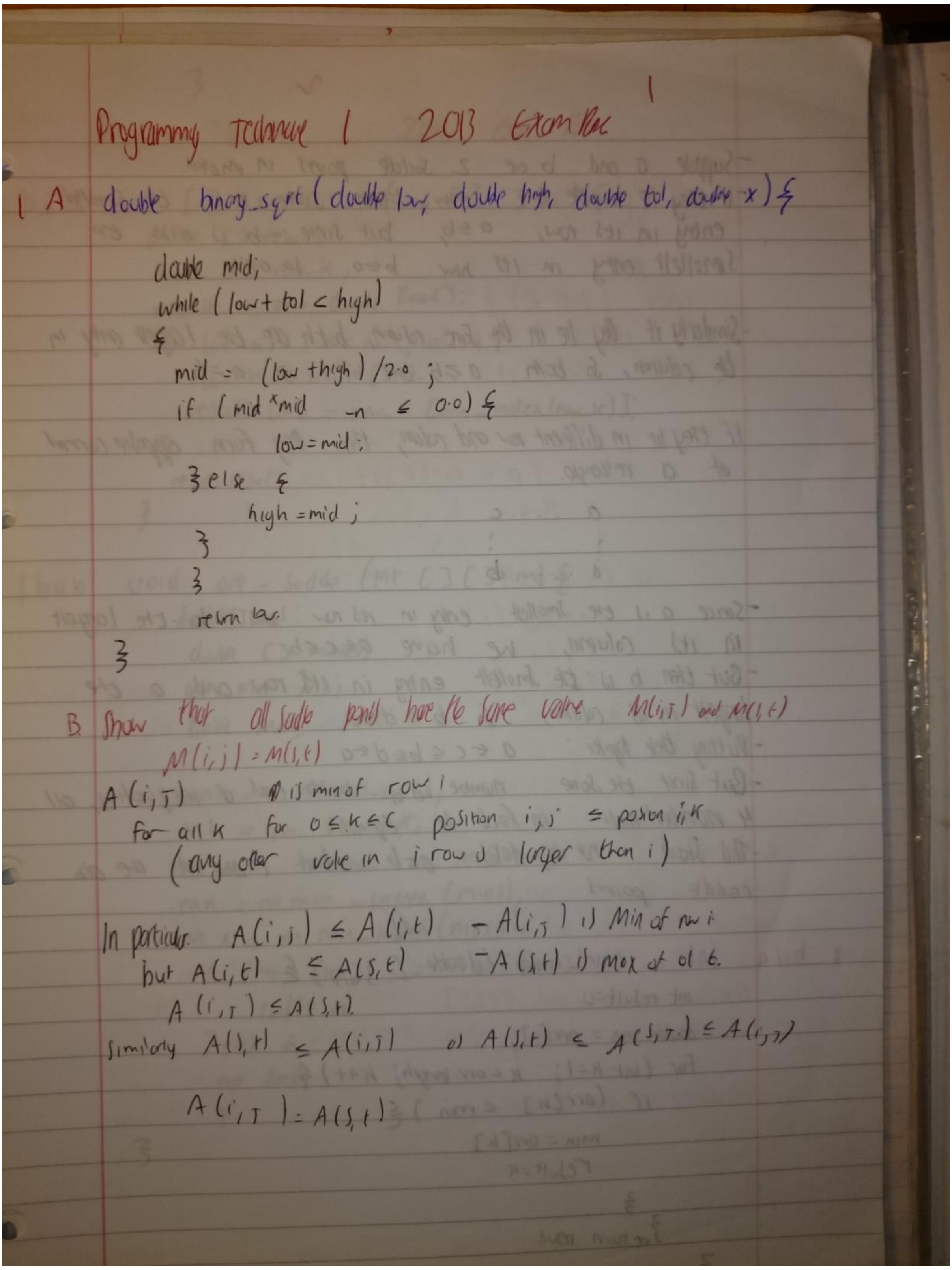


Programming Technique 1 2013 Exam Rec

1 A

```
double binary_sqrt(double low, double high, double tol, double x) {

    double mid;
    while (low + tol < high)
    {
      mid = (low + high) /2.0 ;
      if (mid * mid  -n  ≤ 0.0) {
             low = mid ;
       } else {
            high = mid ;
       }
     }
     return low.
 }
```

B Show that all saddle points have the same value $M(i,j)$ and $M(s,t)$

$M(i,j) = M(s,t)$

$A(i,j)$ ~~~~ is min of row i

for all k for $0 \le k \le C$ position i,j ≤ position i,k

(any other value in i row is larger than i)

In particular $A(i,j) \le A(i,t)$ – $A(i,j)$ is Min of row i

but $A(i,t) \le A(s,t)$ – $A(s,t)$ is max of col t.

$A(i,j) \le A(s,t)$.

Similarly $A(s,t) \le A(i,j)$ as $A(s,t) \le A(s,j) \le A(i,j)$

$A(i,j) = A(s,t)$

- Suppose a and b are 2 saddle points in matrix.
- If they lie in the same row, then since a is the smallest entry in its row, $a \leq b$, but since b is also the smallest entry in its row $b \leq a$ so $a = b$

- Similarly if they lie in the same column, both are the largest entry in the column, so both $a \geq b$ and $b \geq a \Rightarrow a = b$

If they lie in different row and colum, then they form opposite corners of a rectangle.

```
a . . . . c
⋮         ⋮
d . . .   b
```

- Since a is the smallest entry in its row and b the largest in its column, we have $a \leq c \leq b$.
- But then b is the smallest entry in it's row and a the largest in its column, so $b \leq d \leq a$.
- Putting these together: $a \leq c \leq b \leq d \leq a$
- But since the same number lies at both end of inequality, all 4 numbers must in fact be equal.
- This shows not only does $a = b$ but c and d are also saddle points

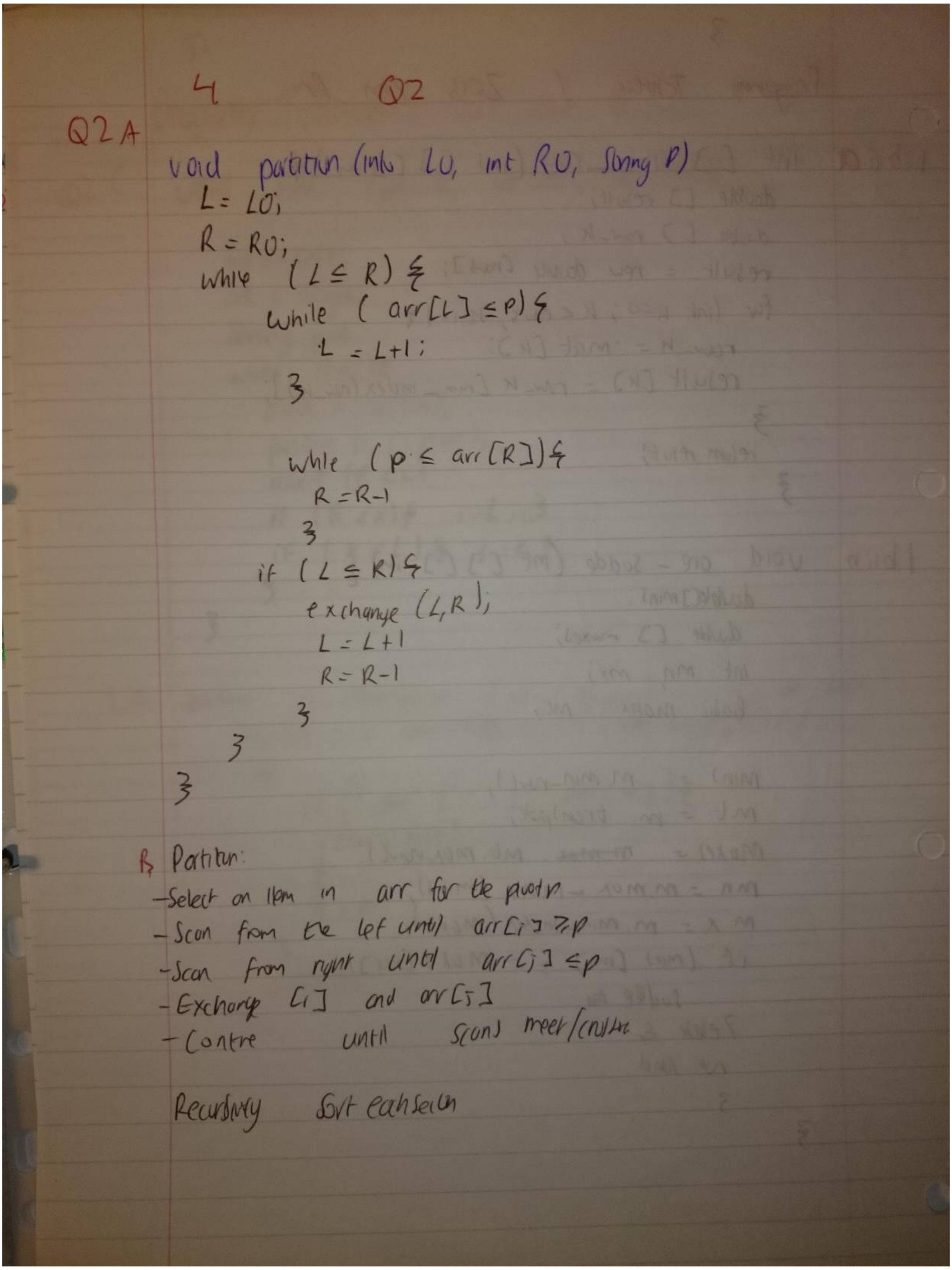
1 biia

```
int - min_index (double [ ]arr) {
    int result=0.
    double min = arr[0];
    For (int k=1;  k<arr.length; k++) {
        if (arr[k] <min ) {
            min = arr[k]
            result=k

        }
    }return resut.
}
```

Programming Techniques 1 2013 Exam Paper

1 b ii a
```
int [] min_row (int [] [] m) {
   double [] result;
   double [] row_k;
   result = new double [rows];
   for (int k=0; k < rows; k++) {
      row_k = mat [k]:
      result [k] = row_k [min_index(row_k)];
   }
   return result;
}
```

1 b ii b
```
void one_saddle (int [] [] m) {
   double[] mins:
   double [] maxes;
   int mn, mx;
   Basic matrix   mt;

   mins = m.min_row();
   mt = m. transpose;
   maxes = ~~m.max~~. mt.max_row():
   mn = m.max_index (mins),
   mx = m.min_index (maxs);
   if (mins [mn] == maxs [mx])
        saddle pt
   } else {
   not found
   }
}
```



4 Q2

Q2A

```
void partition (int LO, int RO, String P)
  L = LO;
  R = RO;
  while (L ≤ R) {
      while ( arr[L] ≤ p) {
          L = L+1;
      }

      while (p ≤ arr[R]) {
          R = R-1
      }
      if (L ≤ R) {
          exchange (L,R);
          L = L+1
          R = R-1
      }
  }
}
```

B Partition:

- Select an item in arr for the pivot p
- Scan from the lef until arr[i] ≥ p
- Scan from right until arr[j] ≤ p
- Exchange [i] and arr[j]
- Contine until scans meet/cross

Recursively sort each section

Programmy Technicel 203 Exam Q

Q2 C
```
void find (int k, int left, int right){

    int i =left;
    int j =right;
    String pivot.
    while (i<j){
       pivot = arr[k];
       partition (i, j, pivot);
       assert (R<L)
       if (R<k){ i=L; }
       if ( k<L){ j=R }
     }
}
```

3 Ai:
```
boolean is_derangement (int [] der)          k = der.length
   for (int i = ~~der.length~~ 0; i<der.length, i++)
     {
        if (! der [i] == der[k] ){
         return false)
        }
        k = k-1
     }
    return true
  }
```