Chap 2. Arrays and Structures

Array

- ADT 2.1: abstract data type Array
- Arrays in C
- e.g.,

```
int list[5], *plist[5];
// base address (α) = address of list[0]
// address of list[i] = α+ i*sizeof(int)
int *list1;
int list2[5];
// list2: pointer to list2[0]
// list2+i: pointer to list2[i]
// (list2+i) = &list2[i] and *(list2+i) = list2[i]
```

- Two-dimensional array
  - e.g., int x[3][5]
  - Figure 2.2: array of arrays representation

Polynomials

- ADT 2.2: abstract data type Polynomial
- Two types of polynomial representation
  - § 2.4.2
  - Example
    - $A(x) = 2x^{1000} + 1$
    - $B(x) = x^4 + 10x^3 + 3x^2 + 1$
  - Figure 2.3: array representation of two polynomials
- Polynomial addition
  - Program 2.5: initial version of padd function
    - Representation-independent (§ 2.4.2 first paragraph)
  - Program 2.6: function to add two polynomials
  - Program 2.7: function to add a new term

Sparse matrix

- Figure 2.4: two matrices
- ADT 2.3: abstract data type SparseMatrix
- Sparse matrix representation
  - SparseMatrix Create(maxRow, maxCol) ::= // § 2.5.2
- Figure 2.5: sparse matrix and its transpose stored as triples
- Program 2.8: transpose of a sparse matrix
- Program 2.9: fast transpose of a sparse matrix