- We have represented the two polynomials $A(x) = 2x^{1000}+1$ and $B(x) = x^4+10x^3+3x^2+1$ in the array terms. The index of the first term of A and B is given by $\boxed{\text{startA and startB}}$, respectively, while $\boxed{\text{finishA and finishB}}$ give the index of the last term of A and B.
- The index of the next free location in the array is given by avail.
- To use $A(x)$ we must pass in startA and finishA. Any polynomial A that has n nonzero terms has startA and finishA such that finishA = startA + (n-1).

Comparison of the two representations

- In the current representation, the problem of many 0 terms is solved
- $A(x) = 2x^{1000}+1$ uses only 6 units of storage: one for startA, one for finishA, two for the coefficients, and two for the exponents.
- When all the terms are nonzero, the current representation requires about twice as much space as the first one. Unless we know beforehand that each of our polynomials has few zero terms, our current representation is probably better.

2.4.3 Polynomial Addition

```
void padd(int startA, int finishA, int startB, int finishB, int *startD, int *finishD)
{ /* add A(x) and B(x) to obtain D(x) */
   float coefficient; *startD = avail;
   while (startA <= finishA && startB <= finishB)
     switch (COMPARE(terms[startA].expon, terms[startB].expon)) {
       case -1: /* a expon < b expon
         attach(terms[startB].coef, terms[startB].expon);
         startB++;
         break;
       case 0: /* equal exponents */
         coefficient = terms[startA].coef + terms[startB].coef;
```