$0 \leq i < j \wedge j = n$

$\langle \Sigma j : j = n : \langle \Sigma i : 0 \leq i < j : (A_i - A_j)^2 \rangle \rangle$

$= \langle \Sigma i : 0 \leq i < n : (A_i - A_n)^2 \rangle$

$(A_i)^2 - 2 \cdot A_i \cdot A_n + (A_n)^2$

Programming Languages: Imperative Program Construction Final

Shin-Cheng Mu

Autumn Term, 2024

$n \geq 1$

Notes before the exam:

- As mentioned in the class, while the solutions to practicals try to explain to you how one can came up with the invariants (and thus do not do all the necessary proofs), in this exam you do not need to do so. Instead, present the final program and *relatively detailed* proofs of invariant initialization, invariant preservation, termination, etc.. The proofs do not need to be as detailed before, but try to do what a 30+ point question's worth of proofs.
- The exam consists of 4 problems worth 120 points in total — you can see the final 20 points as bonus points.

---

1. (40 total points) The aim of this problem is to derive a quick way to compute $\langle \Sigma i\, j : 0 \leqslant i<j<N : (A[i] - A[j])^2 \rangle$ for array A and constant N.

   (a) (10 points) Take n as a constant such that $1 \leqslant n$. Show that (complete the missing parts in ...)

   $j = n$

   $\langle \Sigma i\, j : 0 \leqslant i < j < n+1 : (A[i] - A[j])^2 \rangle =$
   $\quad \dots \langle \Sigma i\, j : 0 \leqslant i < j < n : (A[i] - A[j])^2 \rangle \dots$
   $\quad \dots \langle \Sigma i : 0 \leqslant i < n : A[i]^2 \rangle \dots \langle \Sigma i : 0 \leqslant i < n : A[i] \rangle \dots$.

   $n \geq 1 \quad n = 0$?

   Hint:
   - If the expression gets too long, you may give names to some sub-expressions in the comments.
   - You may need the following properties, for $\oplus$ that is associative, commutive, with identity e, and for $\otimes$ such that $(x \oplus y) \otimes z = (x \otimes z) \oplus (y \otimes z)$:

   $$\langle \oplus i : R : F \rangle \oplus \langle \oplus i : R : G \rangle = \langle \oplus i : R : F \oplus G \rangle, \quad (1)$$
   $$\langle \oplus i : R : F \otimes G \rangle = \langle \oplus i : R : F \rangle \otimes G, \text{ where } i \text{ is not in } G, \quad (2)$$
   $$\langle \Sigma i : 0 \leqslant i < n : K \rangle = n \times K \text{ for constant } K. \quad (3)$$

   (b) (30 points) Construct a program for the following spec:

   con $N : Int\{N \geqslant 1\}$; A : **array** $[0..N)$ **of** Int;
   var $r : Int$;
   S
   $\{r = \langle \Sigma i\, j : 0 \leqslant i < j < N : (A[i] - A[j])^2 \rangle\}$.

   The answer should consist of the program and its correctness proof. The derivation (how you come up with the program) can be omitted, however. You get full points only if you come up with a linear-time program. Routine proofs (e.g. termination) can be very brief, so you can focus on the more tricky parts. **Hint:**
   - When you split off a value, be sure that the range is non-empty.
   - You may need to introduce two more variables.

$A\ 18 = 2 \times A\ (39) = 2\ (39 + A(38))$

- You will need the result of the previous subproblem. Proofs given in there need not be repeated.

2. (30 points) The function A below is defined on natural numbers by:

$$
\begin{aligned}
&A\ 0 = 1 \\
&A\ n = 2 \times A\ (n\ /\ 2) \quad \text{, if } \textit{even } n \text{ and } 0 < n, \\
&A\ n = n + A\ (n - 1) \quad \text{, if } \textit{odd } n.
\end{aligned}
$$

$0\ A\ 18 = A\ 78$

~~$18 + A(17)$~~

$r + k \times A\ n = A\ N$

Construct a program computing $A\ N$:

con $N : \mathit{Int}\ \{0 \leqslant N\}$
var $r : \mathit{Int}$
?
$\{r = A\ N\}$

The answer should consist of the program and its correctness proof. The derivation can be omitted. To get full score, you should derive a program that uses a single loop.

Hint:

- To build the invariant you will need to introduce more variables. Use an invariant of the form $A\ N = ...$, such that under boundary cases, the righthand-side of $A\ N = ...$ is trivial to compute. Assign the result to r as the last step of the program.
- To find out what "..." could be, try to expand A (for example, try $A\ 78$) and see whether there is a pattern.

3. (30 total points) A sequence of n numbers is *bitonic* if it has the form $x_0 < x_1 ... < x_p \geqslant x_{p+1} \geqslant ... x_{n-1}$. It is a concatenation of an increasing sequence and a descending sequence, either of which can be empty. For example, $xs = [2, 4, 5, 5, 3, 1]$ is a bitonic sequence. With index starting from 0, the index of the "highest point" of xs, also called the *bitonic point*, is 2 (having value 5). The lists $[2, 3, 4]$ and $[4, 3, 2]$ are bitonic too, respectively having bitonic points at index 2 and 0. For an counter example, $[2, 4, 5, 5, 3, 4]$ is not bitonic.

Given a *non-empty* array containing a bitonic sequence, the task is to find the bitonic point *using binary search*.

(a) (10 points) The following is an incomplete specification of bitonicity and the problem specificaiton.

$$
\begin{aligned}
&\mathit{bitonic} \quad n = \langle \exists p : 0 \leqslant p < n : \mathit{increasing}\ ??\ \wedge\ \mathit{descending}\ ??\rangle\ , \\
&\mathit{increasing}\ x\ y = \langle \forall i : ? : A[i] < A[i+1]\rangle\ , \\
&\mathit{descending}\ x\ y = \langle \forall i : ? : A[?] \geqslant ?\rangle\ .
\end{aligned}
$$

con $N : \mathit{Int}\ \{1 \leqslant N\}$
con $A :$ **array** $[0..N)$ **of** $\mathit{Int}\ \{\mathit{bitonic}\ N\}$
var $l, m, r : \mathit{Int}$

bitonic_point
$\{???\}$

Upon termination, l should point to the bitonic point. Finish the definition and specification.

(b) (10 points) Recall the outline of binary search:

$\{M < N \wedge \Phi\ M\ N\}$ $0\ N$
$l, r := M, N$ $= [0, N$
$\{\Phi\ l\ r \wedge M \leqslant l < r \leqslant N, \mathit{bnd} : r - l\}$
do $l + 1 \neq r \rightarrow \{... \wedge l + 2 \leqslant r\}$
  $m :=$ anything s.t. $l < m < r$
  $\{... \wedge l < m < r\}$
  if $\Phi\ m\ r \rightarrow l := m$
  $\mid \Phi\ l\ m \rightarrow r := m$
  fi
od
$\{M \leqslant l < N \wedge \Phi\ l\ (l+1)\}$ r $\quad l = p$

e.g. $M = 0,\ N = N$
for $[0 ... N)$ array

$A(l-1) < A(l) \geq A(l+1)$

λ m

Define Φ for this problem of finding the bitonic point. Argue why *bitonic* $N \wedge \Phi\ l\ r \wedge 0 \leqslant l < m < r \leqslant N \Rightarrow \Phi\ l\ m \vee \Phi\ m\ r$ holds. **Note**: for this question you do not have to be as formal as we usually were. Short, informal arguments mixing equational reasoning and verbal explanation will suffice.

(c) (10 points) Present the completed program that finds the bitonic point. For this question, you do not need to show me any proofs — but you still have to make sure that the program is correct by our standard. **Hint:** apparently you cannot just use $\Phi\ m\ r$ and $\Phi\ l\ m$ in the code. You will need to simplify them, but be careful about what they get simplified to!

4. (20 bonus points) Prove the correctness of the following program (where the ... parts are routine and omitted).

con $N : Int\ \{N \geqslant 0\}$
con $X :$ **array** $[0..N)$ **of** $Int\ \{\langle \forall i : 0 \leqslant i < N : 0 \leqslant X[i] < 100 \rangle\}$
var $h :$ **array** $[0..100)$ **of** Int

...
$\{\langle \forall i : 0 \leqslant i < 100 : h[i] = \infty \rangle\}$
$n := N$
do $n \neq 0 \rightarrow$
  $h[X[n-1]] := n - 1$
  $n := n - 1$
od
$\{\langle \forall i : 0 \leqslant i < 100 : h[i] = \langle \downarrow k : 0 \leqslant k < N \wedge X[k] = i : k \rangle \rangle\}$.

Again, routine proofs can be discharged briefly so you can focus on the more tricky parts. Hints:

- To prove $P \Rightarrow Q$, you may try to prove Q while assuming P to be true.
- Similarly, to prove $\langle \forall i : P : Q \rangle$, you may try to prove Q while assuming P to be true.
- Properties regarding a function defined by cases ($f = xxx$ if P holds, $f = yyy$ otherwise) can be proved by discussing these cases separately.
- The following property may be useful.

$$\langle \downarrow k : x \leqslant k < y \wedge P : k \rangle \downarrow z = z \ \ , \text{if } z < x. \qquad (4)$$