```
A_pos = malloc((m+1) * sizeof(int));
A_crd = malloc(A_crd_size * sizeof(int));

row = malloc(row_size * sizeof(bool));
memset(row, 0, row_size * sizeof(bool));
rowlist = malloc(row_size);

A_pos[0] = 0;
for (int i = 0; i < m; i++) {
  rowlist_size = 0;
  for (int pB = B_pos[i]; pB < B_pos[i+1]; pB++) {
    int k = B_crd[pB];
    for (int pC = C_pos[k]; pC < C_pos[k+1]; pC++) {
      int j = C_crd[pC];
      if (!row[j]) {
        rowlist[rowlist_size++] = j;
        row[j] = true;
      }
    }
  }

  // Sort row indices
  sort(rowlist, rowlist_size);

  // Make sure A_crd is large enough
  if (A_crd_size < (A_pos[i] + rowlist_size)) {
    A_crd_size = (A_pos[i] + rowlist_size) * 2;
    A_crd = realloc(A_crd, A_crd_size * sizeof(int));
  }

  // Copy row workspace indices to A_crd
  for (int prow = 0; prow < rowlist_size; prow++) {
    int j = rowlist[prow];
    A_crd[A_pos[i] + prow] = j;
    row[j] = false;
  }
  A_pos[i+1] = A_pos[i] + rowlist_size;
}
A = malloc(A_pos[m] * sizeof(float));
```