```
(* This notebook estimates the total FLOPS and Storage needed for the 2-to-1 and
  4-to-1 merge algorithms for the different implementations of the HPS method *)
(* 2-to-1 Vertical Merge *)
ARows = 6 M;
ACols = 6 M;
BRows = 6 M;
BCols = M;
CRows = M;
CCols = 6 M;
DRows = M;
DCols = M;
SRows = M;
SCols = 6 M;
TRows = 6 M;
TCols = 6 M;
FLOPS4SxVertical = DRows^3;
FLOPS4TxVertical = BRows * BCols * SCols;
STORAGE4SxVertical = SRows * SCols;
STORAGE4TxVertical = TRows * TCols;
(* 2-to-1 Horizontal Merge *)
ARows = 8 M;
ACols = 8 M;
BRows = 8 M;
BCols = 2 M;
CRows = 2 M;
CCols = 8 M;
DRows = 2 M;
DCols = 2 M;
SRows = 2 M;
SCols = 8 M;
TRows = 8 M;
TCols = 8 M;
FLOPS4SxHorizontal = DRows^3;
FLOPS4TxHorizontal = BRows * BCols * SCols;
STORAGE4SxHorizontal = SRows * SCols;
STORAGE4TxHorizontal = TRows * TCols;
(* 2-to-1 FLOPS *)
VerticalFLOPS = 2 * (FLOPS4SxVertical + FLOPS4TxVertical);
HorizontalFLOPS = FLOPS4SxHorizontal + FLOPS4TxHorizontal;
TotalFLOPSx2to1 = VerticalFLOPS + HorizontalFLOPS;
(* 2-to-1 Storage *)
TotalStoragex2to1 = 2 * (STORAGE4SxVertical + STORAGE4TxVertical) +
   STORAGE4SxHorizontal + STORAGE4TxHorizontal;
```

```
(* 4-to-1 Merge *)
ARows = 8 M;
ACols = 8 M;
BRows = 8 M;
BCols = 4 M;
CRows = 4 M;
CCols = 8 M;
DRows = 4 M;
DCols = 4 M;
SRows = 4 M;
SCols = 8 M;
TRows = 8 M;
TCols = 8 M;
FLOPS4S = DRows^3;
FLOPS4T = BRows * BCols * SCols;
STORAGE4S = SRows * SCols;
STORAGE4T = TRows * TCols;
TotalFLOPSx4to1 = FLOPS4S + FLOPS4T;
TotalStoragex4to1 = STORAGE4S + STORAGE4T;
```

```
In[329]:= Print["2-to-1 Total FLOPS: ", TotalFLOPSx2to1]
          Print["4-to-1 Total FLOPS: ", TotalFLOPSx4to1]
          Print["2-to-1 Total Storage: ", TotalStoragex2to1]
          Print["4-to-1 Total Storage: ", TotalStoragex4to1]
```

2-to-1 Total FLOPS: 210 M^3

4-to-1 Total FLOPS: 320 M^3

2-to-1 Total Storage: 164 M^2

4-to-1 Total Storage: 96 M^2