for (x = 0; x < sequenceX.length; x++)
{
    H_up = readFromGlobal(x); F_up = readFromGlobal(x);
    sequenceXElement = readFromTexture(x);
    H_upleft = H_init; H_init = H_up;
    unsigned int back = 0;

    // |
    // V
    for(i = 0; i < 8; i++)
    {
        H_left = readFromShared(i); E_left = readFromShared(i);
        sequenceYElement = readFromShared(i);

        //reading value of substitution matrix from constant memory
        similarity = readFromConstant(sequenceXElement, sequenceYElement);

        E_current = max(E_left - gapEx, H_left - gapOp);
        F_current = max(F_up - gapEx, H_up - gapOp);

        H_current = max(E_current, F_current);
        H_current = max(H_current, H_upleft + similarity);

        //backtracking arrays
        back <<= 1;
        back |= (H_current == F_current) ||
                (H_current == H_upleft + similarity); //if go up
        back <<= 1;
        back |= (H_current == E_current) &&
                (H_current != F_current) ||
                (H_current == H_upleft + similarity); //if go left
        back <<= 1;
        back |= F_current == F_up - gapEx; //if continue up
        back <<= 1;
        back |= E_current == E_left - gapEx; //if continue left

        //initialize variables for next iteration
        writeToShared(H_current, i); writeToShared(E_current, i);
        H_upleft = H_left; H_up = H_current; F_up = F_current;
    }
    writeToGlobal(H_up,x); writeToGlobal(F_up,x); writeToGlobal(back,x,y);
}