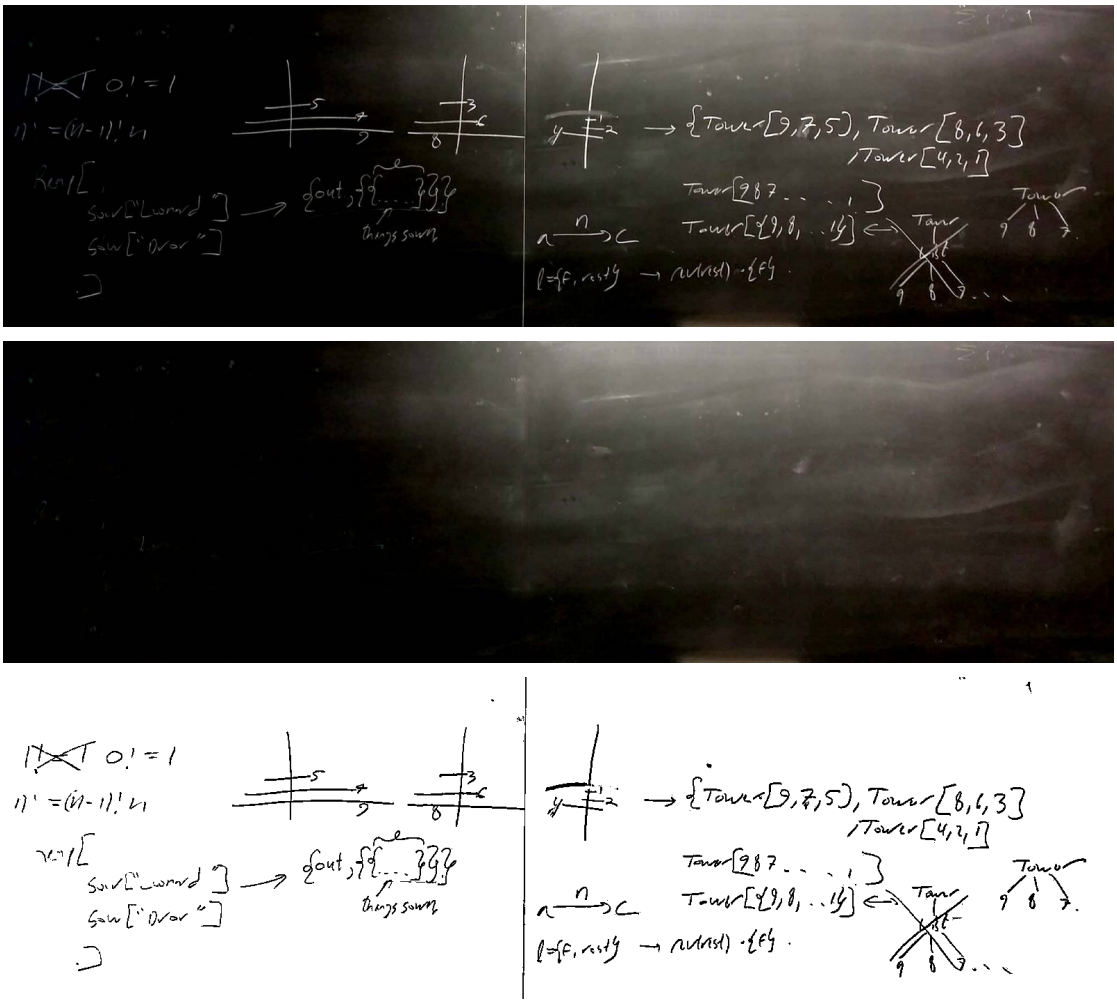
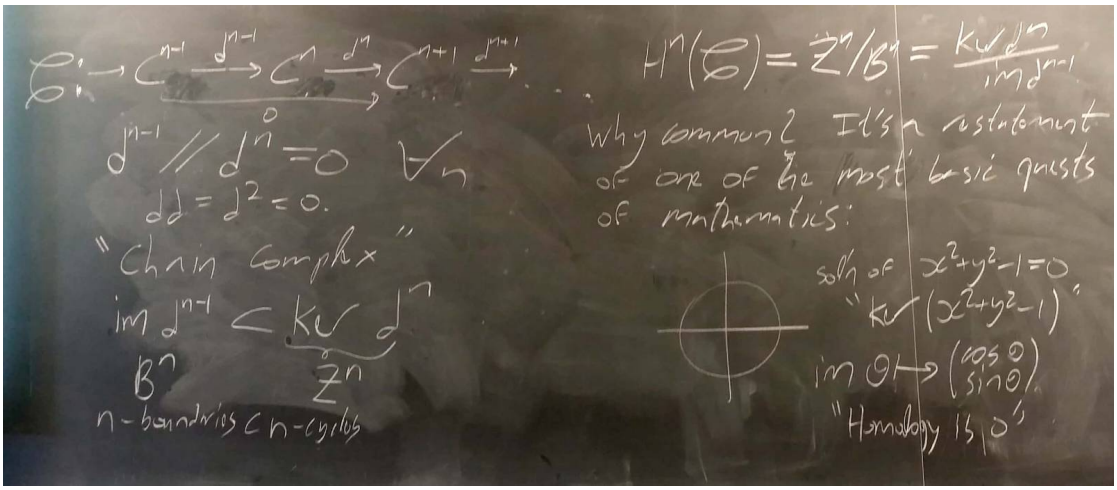
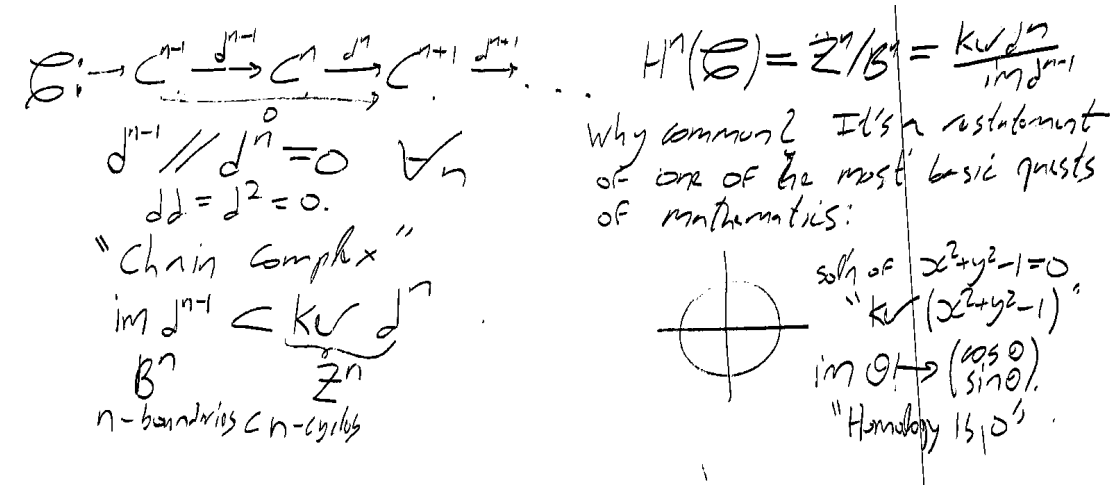
```
(* ImgImprove works by finding the edges of the image,
which shows us where the characters are. Inpainting is then used to estimate
 what the background looks like without any text. This is subtracted from the
 original image to obtain only the text (thus eliminating lighting effects),
which is then sharpened (essentially a deconvolution with a Gaussian kernel),
binarized, and colour flipped. The variable t defines the
  threshold used for binarizing. A value of t = 0.2 seems to work well*)

ImgImprove[img_, t_] :=
 Module[{edges, background, out}, edges = Dilation[EdgeDetect[img, 0.5], 3];
  background = Inpaint[img, edges] ;
  ;
  out = ColorNegate[Binarize[Sharpen[ImageSubtract[img, background], 10], t]];
  Print[img];
  Print[background]; Print[out]]


(* Load 5 sample images*)

img1 = ;

img2 = ;

img3 = ;

img4 = ;

img5 = ;

ImgImprove[img1, 0.2]
```

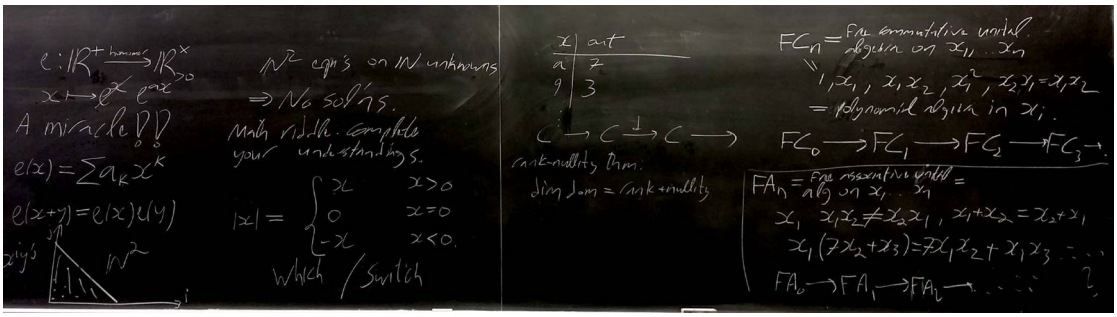


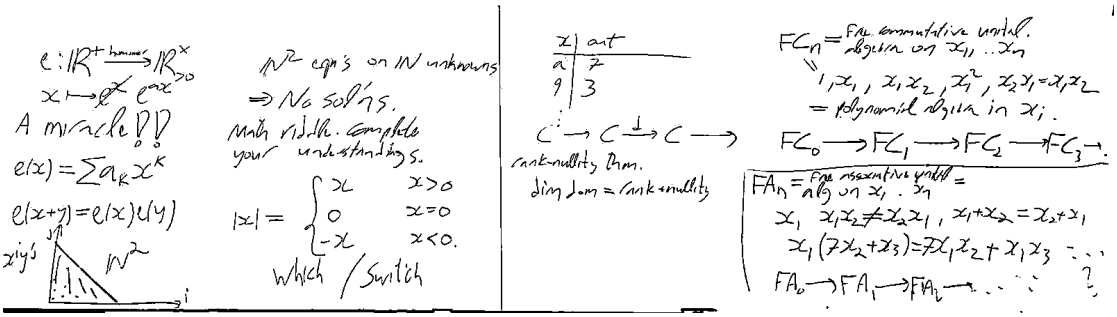
ImgImprove[img2, 0.2]



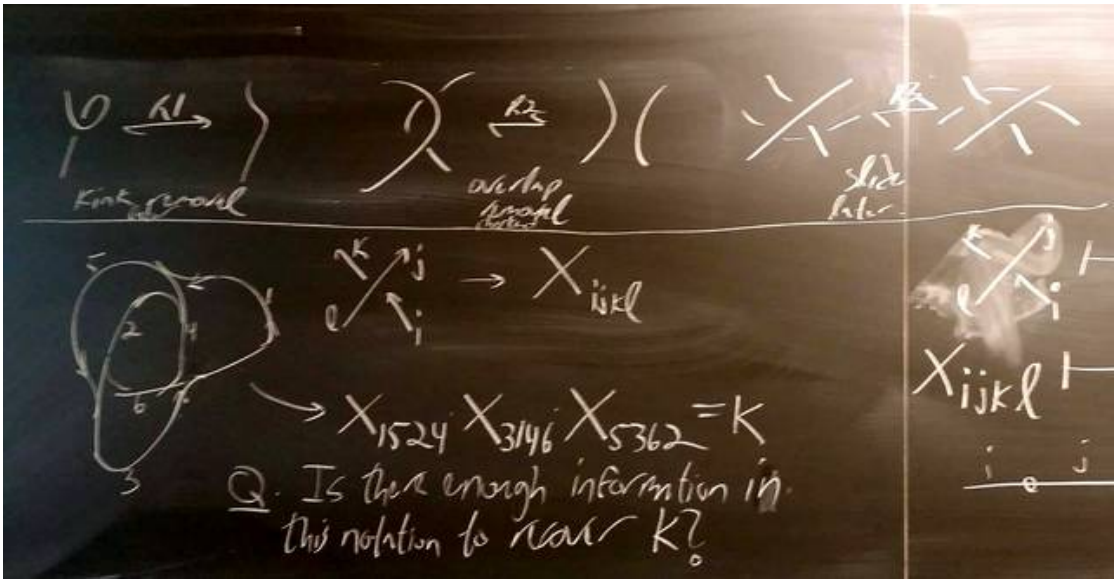


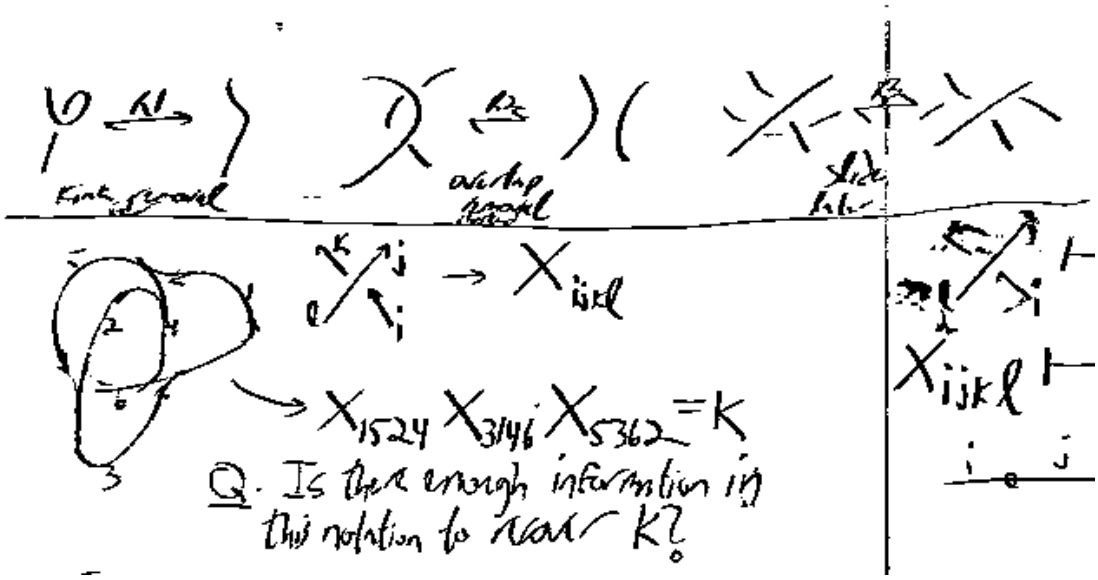

```
ImgImprove[img3, 0.2]
```



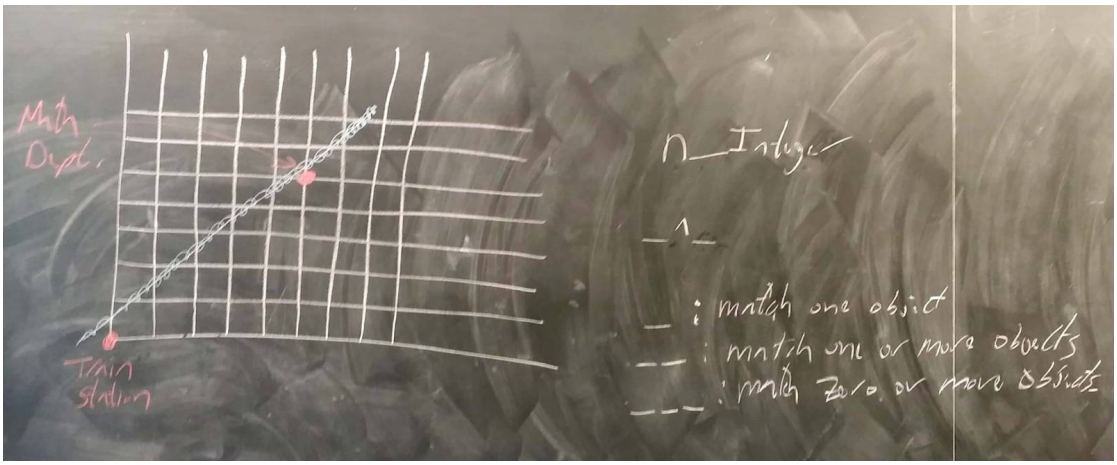


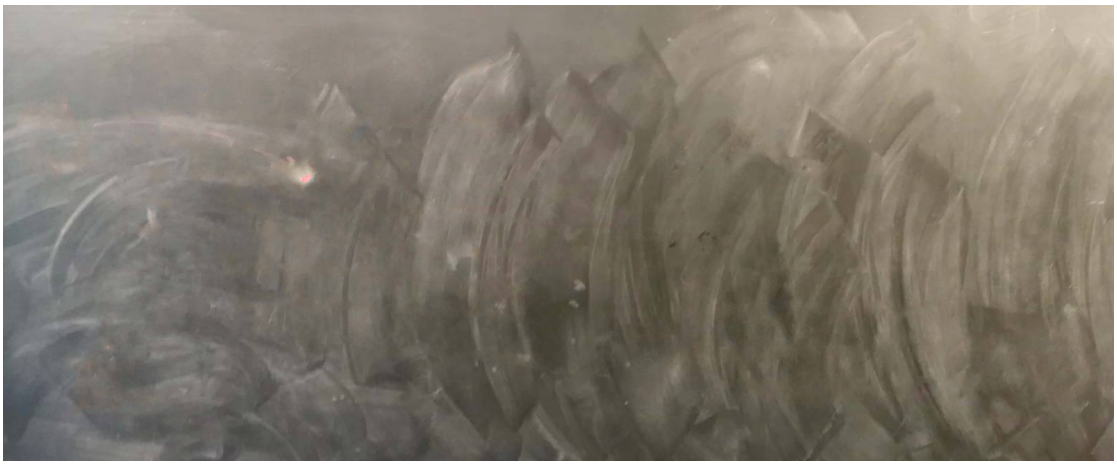
```
ImgImprove[img4, 0.3]
```

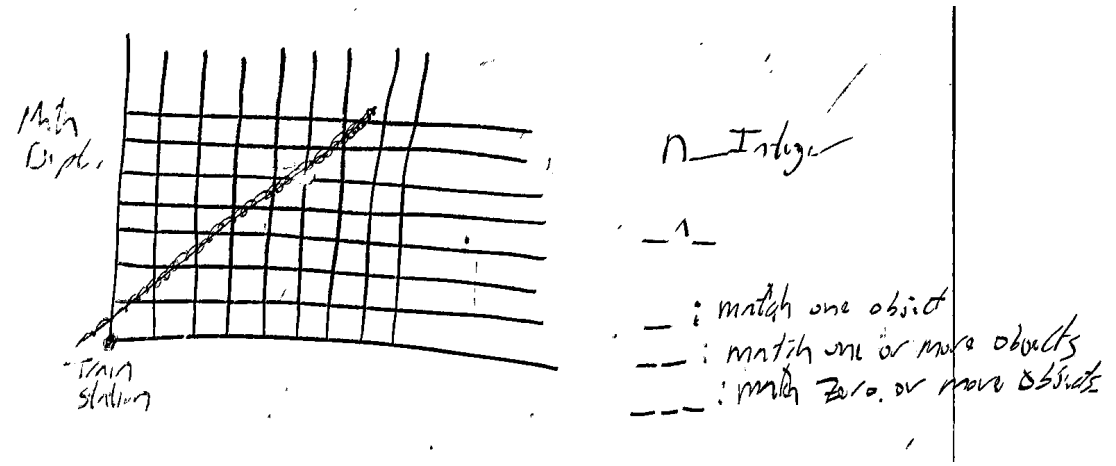





```
ImgImprove[img5, 0.2]
```


Math Dept.
Train station
n_Integer
_^_
_ : match one object
__ : match one or more objects
___ : match zero or more objects




Math Dept.
Train station
n_Integer
_^_
_ : match one object
__ : match one or more objects
___ : match zero or more objects